

ShearLab 3D: Faithful Digital Shearlet Transforms based on Compactly Supported Shearlets

Gitta Kutyniok, Technische Universität Berlin
 Wang-Q Lim, Technische Universität Berlin
 Rafael Reisenhofer, Technische Universität Berlin

Wavelets and their associated transforms are highly efficient when approximating and analyzing one-dimensional signals. However, multivariate signals such as images or videos typically exhibit curvilinear singularities, which wavelets are provably deficient of sparsely approximating and also of analyzing in the sense of, for instance, detecting their direction. Shearlets are a directional representation system extending the wavelet framework, which overcomes those deficiencies. Similar to wavelets, shearlets allow a faithful implementation and fast associated transforms. In this paper, we will introduce a comprehensive carefully documented software package coined ShearLab 3D (www.ShearLab.org) and discuss its algorithmic details. This package provides MATLAB code for a novel faithful algorithmic realization of the 2D and 3D shearlet transform (and their inverses) associated with compactly supported universal shearlet systems incorporating the option of using CUDA. We will present extensive numerical experiments in 2D and 3D concerning denoising, inpainting, and feature extraction, comparing the performance of ShearLab 3D with similar transform-based algorithms such as curvelets, contourlets, or surfacelets. In the spirit of reproducible research, all scripts are accessible on www.ShearLab.org.

Categories and Subject Descriptors: G.1.2 [Numerical Analysis]: Wavelets and fractals

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Imaging Sciences, Software Package, Shearlets, Wavelets

ACM Reference Format:

Gitta Kutyniok, Wang-Q Lim, and Rafael Reisenhofer, 2014. ShearLab 3D: Faithful Digital Shearlet Transform based on Compactly Supported Shearlets. *ACM Trans. Math. Softw.* V, N, Article A (January YYYY), 35 pages.

DOI: <http://dx.doi.org/10.1145/0000000.0000000>

1. INTRODUCTION

Wavelets have had a tremendous success in both theoretical and practical applications such as, for instance, in optimal schemes for solving elliptic PDEs or in the compression standard JPEG2000. A wavelet system is based on one or a few generating functions to which isotropic scaling operators and translation operators are applied to. One main advantage of wavelets is their ability to deliver highly sparse approximations of 1D signals exhibiting singularities, which makes them a powerful maximally flexible tool in being applicable to a variety of problems such as denoising or detection of singularities.

This work is supported in part by the Einstein Foundation Berlin, by Deutsche Forschungsgemeinschaft (DFG) Grant SPP-1324 KU 1446/13 and DFG Grant KU 1446/14, by the DFG Collaborative Research Center TRR 109 “Discretization in Geometry and Dynamics”, and by the DFG Research Center MATHEON “Mathematics for key technologies” in Berlin.

Author’s addresses: Gitta Kutyniok, Wang-Q Lim, and Rafael Reisenhofer, Department of Mathematics, Technische Universität Berlin, 10623 Berlin, Germany.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© YYYY ACM 0098-3500/YYYY/01-ARTA \$15.00

DOI: <http://dx.doi.org/10.1145/0000000.0000000>

ties. But similarly important for applications is the fact that wavelets admit a faithful digitalization of the continuum domain systems with efficient algorithms for the associated transform computing the respective wavelet coefficients (cf. [Daubechies 1992; Mallat 2008]).

However, each multivariate situation starting with the 2D situation differs significantly from the 1D situation, since now not only (0-dimensional) point singularities, but in addition typically also (1-dimensional) curvilinear singularities appear; one can think of edges in images or shock fronts in transport dominated equations. Unfortunately, wavelets are deficient to adequately handle such data, since they are themselves isotropic – in the sense of not directional based – due to their isotropic scaling matrix. Thus, it was proven in [Candès and Donoho 2004] that wavelets do not provide optimally sparse approximations of 2D functions governed by curvilinear singularities in the sense of the decay rate of the L^2 -error of best N -term approximation. This causes problems for any application requiring sparse expansions such as any imaging methodology based on compressed sensing (cf. [Davenport et al. 2012]). Moreover, being associated with just a scaling and a translation parameter, wavelets are, for instance, also not capable of detecting the orientation of edge-like structures.

1.1. Geometric Multiscale Analysis

These problems were the reason that within applied harmonic analysis the research area of geometric multiscale analysis arose, whose main goal consists in developing representation systems which efficiently capture and sparsely approximate the geometry of objects such as curvilinear singularities of 2D functions. One approach pursued was to introduce a 2D model situation coined *cartoon-like functions* consisting of functions compactly supported on the unit square while being C^2 except for a closed C^2 discontinuity curve. A representation system was referred to as ‘optimally sparsely approximating cartoon-like functions’ provided that it provided the optimally achievable decay rate of the L^2 -error of best N -term approximation.

A first breakthrough could be reported in 2004 when Candès and Donoho introduced the system of *curvelets*, which could be proven to optimally sparsely approximate cartoon-like functions while forming Parseval frames [Candès and Donoho 2004]. Moreover, this system showed a performance superior to wavelets in a variety of applications (see, for instance, [Herrmann et al. 2008; Starck et al. 2010]). However, curvelets suffer from the fact that in addition to a parabolic scaling operator and translation operator, the rotation operator utilized as a means to change the orientation can not be faithfully digitalized; and the implementation could therefore not be made consistent with the continuum domain theory [Candès et al. 2006]. This led to the introduction of *contourlets* by Do and Vetterli [Do and Vetterli 2005], which can be seen as a filterbank approach to the curvelet transform. However, both of these well-known approaches do not exhibit the same advantages wavelets have, namely a unified treatment of the continuum and digital situation, a simple structure such as certain operators applied to very few generating functions, and a theory for compactly supported systems to guarantee high spatial localization.

1.2. Shearlets and Beyond

Shearlets were introduced in 2006 to provide a framework which achieves these goals. These systems are indeed generated by very few functions to which parabolic scaling and translation operators as well as shearing operators to change the orientation are applied [Kutyniok and Labate 2012]. The utilization of shearing operators ensured – due to consistency with the digital lattice – that the continuum and digital realm was treated uniformly in the sense of the continuum theory allowing a faithful implementation (cf. for band-limited generators [Kutyniok et al. 2012b]). A theory for compactly

supported shearlet frames is available [Kittipoom et al. 2012], showing that although presumably Parseval frames can not be derived, still the frame bounds are within a numerically stable range. Moreover, compactly supported shearlets can be shown to optimally sparsely approximating cartoon-like functions [Kutyniok and Lim 2011]. Also, to date a 3D theory is available [Kutyniok et al. 2012a].

Very recently, two extensions of shearlet theory were explored. One first extension is the theory of α -molecules [Grohs et al. 2013]. This approach extends the theory of parabolic molecules introduced in [Grohs and Kutyniok 2014], which provides a framework for systems based on parabolic scaling such as curvelets and shearlets to analyze their sparse approximation properties. α -molecules are a parameter-based framework including systems based on different types of scaling such as, in particular, wavelets and shearlets, with the parameter α measuring the degree of anisotropy. As a sub-family of this general framework so-called α -shearlets (also sometimes called *hybrid shearlets*) were studied in [Kutyniok et al. 2012a] (cf. also [Keiper 2012]), which can be regarded as a parametrized family ranging from wavelets ($\alpha = 2$) to shearlets ($\alpha = 1$). Again, the frame bounds can be controlled and optimal sparse approximation properties – now for a parametrized model situation – are proven, also for compactly supported systems.

A further extension are *universal shearlets*, which allow a different type of scaling (parabolic, etc.) at each scaling level of α -shearlets by setting $\alpha = (\alpha_j)_j$ with j being the scale, thereby achieving maximal flexibility [Genzel and Kutyniok 2014]. This approach has so far been only analyzed for band-limited generators, deriving properties on the sequence $(\alpha_j)_j$ such that the resulting system forms a Parseval frame.

1.3. Contributions

Several implementations of shearlet transforms are available to date, and we refer to Subsections 2.3 and 5.2 for more details. Most of those focus on the (2D) band-limited case. In this situation, a Parseval frame can be achieved providing immediate numerical stability and a straightforward inverse transform. However, from an application point of view, those approaches typically suffer from high complexity, various artifacts, and insufficient spatial localization. The faithful algorithmic realization suggested in [Lim 2010] by one of the authors was the first to focus on compactly supported shearlets, achieving also low complexity by utilizing separable shearlet generators. Interestingly, this approach could then be improved in [Lim 2013] by utilizing non-separable compactly supported shearlet generators. The key idea behind this seemingly unreasonable approach is that the classical band-limited generators – whose Fourier transforms have wedge-like support – leading to Parseval frames can be much better approximated by non-separable compactly supported functions than by separable ones.

ShearLab 3D builds on the approach from [Lim 2013], and extends it in two ways, namely to universal shearlets as well as to the 3D situation. In addition, elaborate numerical experiments comparing ShearLab 3D with the current state-of-the-art algorithms in geometric multiscale analysis are provided, more precisely with the Nonsubsampled Shearlet Transform in 2D & 3D [Easley et al. 2008], the Nonsubsampled Contourlet Transform in 2D [da Cunha et al. 2006], the Fast Discrete Curvelet Transform in 2D [Candès and Donoho 1999a], the Surfacelet Transform in 3D [Do and Lu 2007], and finally also with the classical Stationary Wavelet Transform in 2D. The algorithms were tested with respect to denoising and inpainting in 2D and 3D as well as separation in 2D (separation in 3D requires additional systems which renders a comparison unfair). Each time carefully chosen performance measures were specified and served as an objective basis for the comparisons. It could be shown that indeed ShearLab 3D outperforms the other algorithms in most tasks, also concerning speed. It is interesting to note that with respect to video denoising ShearLab 3D as a universally applicable

tool is only marginally beaten by the specifically for this task designed BM3D algorithm [Maggioni et al. 2012].

In the spirit of reproducible research [Donoho et al. 2009], ShearLab 3D as well as the codes for all comparisons are freely accessible on www.ShearLab.org.

1.4. Outline

This paper is organized as follows. We first review the main definitions and results from frame and shearlet theory in Section 2. Section 3 is then devoted to a detailed discussion of the faithful algorithmic realization of the shearlet transform (and its inverse), which is implemented in ShearLab 3D. After a short review of the wavelet transform in Subsection 3.1, the 2D (Subsection 3.2) and 3D (Subsection 3.3) algorithms are discussed, each time first in the parabolic and then in the general situation, and finally in Subsection 3.4 the inverse transform is presented. The actual MATLAB implementation in ShearLab 3D is described in Section 4. Finally, elaborate numerical experiments are presented (Section 5) ranging from denoising to video inpainting in Subsections 5.3 to 5.7 each time carefully comparing (Subsection 5.8) ShearLab 3D with the state-of-the-art algorithms described in Subsection 5.2.

2. DISCRETE SHEARLET SYSTEMS

In this section, we will state the main definitions and necessary results about shearlet systems for $L^2(\mathbb{R}^2)$ and $L^2(\mathbb{R}^3)$. We start though with a brief review of frame theory which – with the notion of frames – provides a functional analytic concept to generalize the setting of orthonormal bases. In fact, shearlet systems do not form bases, but require this extended concept.

2.1. Review: Frame Theory

Frame theory is nowadays used when redundant, yet stable expansions are required. A sequence $(\varphi_i)_{i \in I}$ in a Hilbert space \mathcal{H} is called *frame for \mathcal{H}* , if there exist constants $0 < A \leq B < \infty$ such that

$$A\|x\|^2 \leq \sum_{i \in I} |\langle x, \varphi_i \rangle|^2 \leq B\|x\|^2 \quad \text{for all } x \in \mathcal{H}.$$

If $A = B$ is possible, the frame is referred to as *tight*, in case $A = B = 1$ as a *Parseval frame*. One application of frames is the analysis of elements in a Hilbert space, which is achieved by the *analysis operator* given by

$$T : \mathcal{H} \rightarrow \ell_2(I), \quad T(x) = (\langle x, \varphi_i \rangle)_{i \in I}.$$

Reconstruction of each element $x \in \mathcal{H}$ from Tx is possible by the *frame reconstruction formula* given by

$$x = \sum_{i \in I} \langle x, \varphi_i \rangle S^{-1} \varphi_i, \tag{1}$$

where $Sx = \sum_{i \in I} \langle x, \varphi_i \rangle \varphi_i$ is the *frame operator* associated with the frame $(\varphi_i)_{i \in I}$. We remark that for tight frames, the frame operator is just a multiple of the identity, hence easily invertible. For general frames, it might be difficult to use (1) in practise, since inverting S might be numerically unfeasible, in particular if B/A is large. In those cases, the so-called *frame algorithm*, for instance, described in [Christensen 2003], can be applied. Moreover, in [Gröchenig 1993], the *Chebyshev method* and the *conjugate gradient methods* were introduced, which are significantly better adapted to frame theory leading to faster convergence than the frame algorithm.

2.2. Universal Shearlet Systems

We next introduce 2D and 3D universal shearlet systems, each time first discussing the parabolic case and then extending it to the general case. For more information, we refer to [Kutyniok and Labate 2012] and [Genzel and Kutyniok 2014].

2.2.1. 2D Situation. Shearlet systems can be regarded as consisting of certain generating functions whose resolution is changed by a parabolic scaling matrix A_{2^j} or \tilde{A}_{2^j} defined by

$$A_{2^j} = \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix} \quad \text{and} \quad \tilde{A}_{2^j} = \begin{pmatrix} 2^{j/2} & 0 \\ 0 & 2^j \end{pmatrix},$$

whose orientation is changed by a shearing matrix S_k or S_k^T defined by

$$S_k = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix},$$

and whose position is changed by translation. More precisely, a shearlet system – sometimes in this form also referred to as *cone-adapted* due to the fact that it is adapted to a cone-like partition in frequency domain (cf. Figure 1) – is defined as follows.

Definition 2.1. For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$ and $c = (c_1, c_2) \in (\mathbb{R}_+)^2$, the *shearlet system* $SH(\phi, \psi, \tilde{\psi}; c)$ is defined by

$$SH(\phi, \psi, \tilde{\psi}; c) = \Phi(\phi; c_1) \cup \Psi(\psi; c) \cup \tilde{\Psi}(\tilde{\psi}; c),$$

where

$$\begin{aligned} \Phi(\phi; c_1) &= \{\phi_m = \phi(\cdot - c_1 m) : m \in \mathbb{Z}^2\}, \\ \Psi(\psi; c) &= \{\psi_{j,k,m} = 2^{\frac{3}{4}j} \psi(S_k A_{2^j} \cdot - M_c m) : j \geq 0, |k| \leq \lceil 2^{j/2} \rceil, m \in \mathbb{Z}^2\}, \\ \tilde{\Psi}(\tilde{\psi}; c) &= \{\tilde{\psi}_{j,k,m} = 2^{\frac{3}{4}j} \tilde{\psi}(S_k^T \tilde{A}_{2^j} \cdot - \tilde{M}_c m) : j \geq 0, |k| \leq \lceil 2^{j/2} \rceil, m \in \mathbb{Z}^2\}, \end{aligned}$$

with

$$M_c = \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} \quad \text{and} \quad \tilde{M}_c = \begin{pmatrix} c_2 & 0 \\ 0 & c_1 \end{pmatrix}.$$

Its associated transform maps functions to the sequence of shearlet coefficients, hence is merely the associated analysis operator.

Definition 2.2. Set $\Lambda = \mathbb{N}_0 \times \{-\lceil 2^{j/2} \rceil, \dots, \lceil 2^{j/2} \rceil\} \times \mathbb{Z}^2$. Further, let $SH(\phi, \psi, \tilde{\psi}; c)$ be a shearlet system and retain the notions from Definition 2.1. Then the associated *shearlet transform* of $f \in L^2(\mathbb{R}^2)$ is the mapping defined by

$$f \rightarrow \mathcal{SH}_{\phi, \psi, \tilde{\psi}} f(m', (j, k, m), (\tilde{j}, \tilde{k}, \tilde{m})) = (\langle f, \phi_{m'} \rangle, \langle f, \psi_{j,k,m} \rangle, \langle f, \tilde{\psi}_{\tilde{j}, \tilde{k}, \tilde{m}} \rangle),$$

where

$$(m', (j, k, m), (\tilde{j}, \tilde{k}, \tilde{m})) \in \mathbb{Z}^2 \times \Lambda \times \Lambda.$$

A (historically) first class of generators are so-called classical shearlets, which are defined as follows. Let $\psi \in L^2(\mathbb{R}^2)$ be defined by

$$\hat{\psi}(\xi) = \hat{\psi}(\xi_1, \xi_2) = \hat{\psi}_1(\xi_1) \hat{\psi}_2\left(\frac{\xi_2}{\xi_1}\right), \quad (2)$$

where $\psi_1 \in L^2(\mathbb{R})$ is a discrete wavelet in the sense that it satisfies the discrete Calderón condition, given by

$$\sum_{j \in \mathbb{Z}} |\hat{\psi}_1(2^{-j} \xi)|^2 = 1 \quad \text{for a.e. } \xi \in \mathbb{R},$$

with $\hat{\psi}_1 \in C^\infty(\mathbb{R})$ and $\text{supp } \hat{\psi}_1 \subseteq [-\frac{1}{2}, -\frac{1}{16}] \cup [\frac{1}{16}, \frac{1}{2}]$, and $\psi_2 \in L^2(\mathbb{R})$ is a bump function in the sense that

$$\sum_{k=-1}^1 |\hat{\psi}_2(\xi + k)|^2 = 1 \quad \text{for a.e. } \xi \in [-1, 1],$$

satisfying $\hat{\psi}_2 \in C^\infty(\mathbb{R})$ and $\text{supp } \hat{\psi}_2 \subseteq [-1, 1]$. Then ψ is called a *classical shearlet*. With small modifications of the boundary elements, classical shearlets lead to a Parseval frame for $L^2(\mathbb{R}^2)$ [Guo et al. 2006]. The induced tiling of the frequency plane is illustrated in Figure 1.

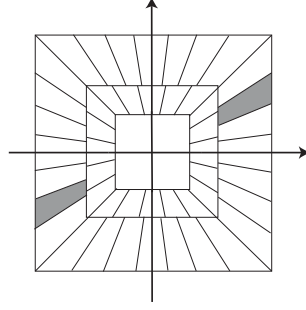


Fig. 1. Tiling of the frequency plane induced by the Parseval frame of classical shearlets.

Some years later, compactly supported shearlets have been studied. It was shown that a large class of compactly supported generators yield shearlet frames with controllable frame bounds [Kittipoom et al. 2012]. One for numerical algorithms particularly interesting special case are separable generators given by $\psi = \psi_1 \otimes \phi_1$, which generate shearlet frames provided that the 1D wavelet function ψ_1 and the 1D scaling function ϕ_1 are sufficiently smooth and ψ_1 has sufficient vanishing moments.

Let us now turn to the more flexible universal shearlets, which were introduced in [Genzel and Kutyniok 2014]. Their definition requires an extension of the scaling matrix to insert a parameter $\alpha \in (0, 2)$ measuring the degree of anisotropy. For this, let $A_{\alpha, 2^j}$ and $\tilde{A}_{\alpha, 2^j}$ be defined by

$$A_{\alpha, 2^j} = \begin{pmatrix} 2^j & 0 \\ 0 & 2^{\alpha j/2} \end{pmatrix} \quad \text{and} \quad \tilde{A}_{\alpha, 2^j} = \begin{pmatrix} 2^{\alpha j/2} & 0 \\ 0 & 2^j \end{pmatrix},$$

A universal shearlet system can then be defined as follows.

Definition 2.3. For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$, $\alpha = (\alpha_j)_j$, $\alpha_j \in (0, 2)$ for each scale j , and $c = (c_1, c_2) \in (\mathbb{R}_+)^2$, the *universal shearlet system* $SH(\phi, \psi, \tilde{\psi}; \alpha, c)$ is defined by

$$SH(\phi, \psi, \tilde{\psi}; \alpha, c) = \Phi(\phi; c_1) \cup \Psi(\psi; \alpha, c) \cup \tilde{\Psi}(\tilde{\psi}; \alpha, c),$$

where

$$\begin{aligned} \Phi(\phi; c_1) &= \{\phi_m = \phi(\cdot - c_1 m) : m \in \mathbb{Z}^2\}, \\ \Psi(\psi; \alpha, c) &= \{\psi_{j,k,m} = 2^{\frac{\alpha_j+1}{4}j} \psi(S_k A_{\alpha_j, 2^j} \cdot - M_c m) : j \geq 0, |k| \leq \lceil 2^{j(\alpha_j-1)/2} \rceil, m \in \mathbb{Z}^2\}, \\ \tilde{\Psi}(\tilde{\psi}; \alpha, c) &= \{\tilde{\psi}_{j,k,m} = 2^{\frac{\alpha_j+1}{4}j} \tilde{\psi}(S_k^T \tilde{A}_{\alpha_j, 2^j} \cdot - \tilde{M}_c m) : j \geq 0, |k| \leq \lceil 2^{j(\alpha_j-1)/2} \rceil, m \in \mathbb{Z}^2\}. \end{aligned}$$

Let us now briefly discuss the situation when all α_j coincide, i.e., $\alpha_0 := \alpha_j$ for all scale j . In this case, $SH(\phi, \psi, \tilde{\psi}; \alpha, c) = SH(\phi, \psi, \tilde{\psi}; c)$ if $\alpha_0 = 1$. Moreover, if $\alpha_0 = 2$ for any scale j , then $SH(\phi, \psi, \tilde{\psi}; \alpha, c)$ becomes an isotropic wavelet system. It should be also mentioned that for $\alpha_0 \rightarrow 0$, the associated universal shearlet system approaches the system of ridgelets [Candès and Donoho 1999b].

The associated transform is then defined similarly as in the parabolic situation.

Definition 2.4. Set $\Lambda = \mathbb{N}_0 \times \{-\lceil 2^{j(\alpha_j-1)/2} \rceil, \dots, \lceil 2^{j(\alpha_j-1)/2} \rceil\} \times \mathbb{Z}^2$. Further, let $SH(\phi, \psi, \tilde{\psi}; \alpha, c)$ be a shearlet system and retain the notions from Definition 2.3. Then the associated *universal shearlet transform* of $f \in L^2(\mathbb{R}^2)$ is the mapping defined by

$$f \rightarrow \mathcal{SH}_{\phi, \psi, \tilde{\psi}} f(m', (j, k, m), (\tilde{j}, \tilde{k}, \tilde{m})) = (\langle f, \phi_{m'} \rangle, \langle f, \psi_{j,k,m} \rangle, \langle f, \tilde{\psi}_{\tilde{j},\tilde{k},\tilde{m}} \rangle),$$

where

$$(m', (j, k, m), (\tilde{j}, \tilde{k}, \tilde{m})) \in \mathbb{Z}^2 \times \Lambda \times \Lambda.$$

In [Genzel and Kutyniok 2014] it was shown that there exists an abundance of scaling sequences $\alpha = (\alpha_j)_j$ such that with a small modification of classical shearlets the system $SH(\phi, \psi, \tilde{\psi}; \alpha, c)$ yields a Parseval frame for $L^2(\mathbb{R}^2)$.

2.2.2. 3D Situation. Turning now to the 3D situation and starting again with the parabolic case, it is apparent that the 2D parabolic scaling matrix A_{2^j} can be extended either by $\text{diag}(2^j, 2^{j/2}, 2^j)$ or $\text{diag}(2^j, 2^{j/2}, 2^{j/2})$. The first case however generates ‘needle-like shearlets’ for which a frame property seems highly unlikely. Hence the second case is typically considered.

For the sake of brevity, we next immediately present the general case of 3D universal shearlets. Then, for $\alpha \in (0, 2)$, we set

$$A_{\alpha, 2^j} = \begin{pmatrix} 2^j & 0 & 0 \\ 0 & 2^{\alpha j/2} & 0 \\ 0 & 0 & 2^{\alpha j/2} \end{pmatrix}, \quad \tilde{A}_{\alpha, 2^j} = \begin{pmatrix} 2^{\alpha j/2} & 0 & 0 \\ 0 & 2^j & 0 \\ 0 & 0 & 2^{\alpha j/2} \end{pmatrix}, \quad \text{and} \quad \check{A}_{\alpha, 2^j} = \begin{pmatrix} 2^{\alpha j/2} & 0 & 0 \\ 0 & 2^{\alpha j/2} & 0 \\ 0 & 0 & 2^j \end{pmatrix}.$$

The shearing matrices are now associated with a parameter $k = (k_1, k_2) \in \mathbb{Z}^2$ and defined by

$$S_k = \begin{pmatrix} 1 & k_1 & k_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \tilde{S}_k = \begin{pmatrix} 1 & 0 & 0 \\ k_1 & 1 & k_2 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{and} \quad \check{S}_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ k_1 & k_2 & 1 \end{pmatrix}.$$

Finally, for $c_1, c_2 \in \mathbb{R}_+$, translation will be given by $M_c = \text{diag}(c_1, c_2, c_2)$, $\tilde{M}_c = \text{diag}(c_2, c_1, c_2)$, and $\check{M}_c = \text{diag}(c_2, c_2, c_1)$. With these notions, a 3D universal shearlet system can then be defined as follows.

Definition 2.5. For $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$, $\alpha = (\alpha_j)_j$, $\alpha_j \in (0, 2)$ for each scale j , and $c = (c_1, c_2) \in (\mathbb{R}_+)^2$, the *universal shearlet system* $SH(\phi, \psi, \tilde{\psi}; \alpha, c)$ is defined by

$$SH(\phi, \psi, \tilde{\psi}; \alpha, c) = \Phi(\phi; c_1) \cup \Psi(\psi; \alpha, c) \cup \tilde{\Psi}(\tilde{\psi}; \alpha, c) \cup \check{\Psi}(\check{\psi}; \alpha, c),$$

where

$$\begin{aligned} \Phi(\phi; c_1) &= \{\phi_m = \phi(\cdot - c_1 m) : m \in \mathbb{Z}^2\}, \\ \Psi(\psi; \alpha, c) &= \{\psi_{j,k,m} = 2^{\frac{\alpha_j+1}{4}j} \psi(S_k A_{\alpha_j, 2^j} \cdot - M_c m) : j \geq 0, |k| \leq \lceil 2^{j(\alpha_j-1)/2} \rceil, m \in \mathbb{Z}^2\}, \\ \tilde{\Psi}(\tilde{\psi}; \alpha, c) &= \{\tilde{\psi}_{j,k,m} = 2^{\frac{\alpha_j+1}{4}j} \tilde{\psi}(S_k^T \tilde{A}_{\alpha_j, 2^j} \cdot - \tilde{M}_c m) : j \geq 0, |k| \leq \lceil 2^{j(\alpha_j-1)/2} \rceil, m \in \mathbb{Z}^2\}, \\ \check{\Psi}(\check{\psi}; \alpha, c) &= \{\check{\psi}_{j,k,m} = 2^{\frac{\alpha_j+1}{4}j} \check{\psi}(S_k^T \check{A}_{\alpha_j, 2^j} \cdot - \check{M}_c m) : j \geq 0, |k| \leq \lceil 2^{j(\alpha_j-1)/2} \rceil, m \in \mathbb{Z}^2\}. \end{aligned}$$

The associated transform is then defined as follows.

Definition 2.6. Set $\Lambda = \mathbb{N}_0 \times \{-[2^{j(\alpha_j-1)/2}], \dots, [2^{j(\alpha_j-1)/2}]\}^2 \times \mathbb{Z}^3$. Further, let $SH(\phi, \psi, \tilde{\psi}, \check{\psi}; \alpha, c)$ be a shearlet system and retain the notions from Definition 2.5. Then the associated *universal shearlet transform* of $f \in L^2(\mathbb{R}^2)$ is the mapping, which maps f to

$$\mathcal{SH}_{\phi, \psi, \tilde{\psi}, \check{\psi}} f(m', (j, k, m), (\tilde{j}, \tilde{k}, \tilde{m}), (\check{j}, \check{k}, \check{m})) = (\langle f, \phi_{m'} \rangle, \langle f, \psi_{j,k,m} \rangle, \langle f, \tilde{\psi}_{\tilde{j},\tilde{k},\tilde{m}} \rangle, \langle f, \check{\psi}_{\check{j},\check{k},\check{m}} \rangle),$$

where

$$(m', (j, k, m), (\tilde{j}, \tilde{k}, \tilde{m}), (\check{j}, \check{k}, \check{m})) \in \mathbb{Z}^3 \times \Lambda \times \Lambda \times \Lambda.$$

In the situation of parabolic scaling it was shown in [Kutyniok et al. 2012a] that a large class of compactly supported generators yields shearlet frames with controllable frame bounds. This is in particular the case for separable generators $\psi = \psi_1 \otimes \phi_1 \otimes \tilde{\phi}_1$ for which ψ_1 is a 1D wavelets with sufficiently many vanishing moments and $\phi_1, \tilde{\phi}_1$ are sufficiently smooth 1D scaling functions. Still in the parabolic case, the situation of band-limited (classical) 3D shearlets was studied in [Guo and Labate 2012], which – similar to the 2D situation – form a Parseval frame for $L^2(\mathbb{R}^3)$ with a small modification of the elements near the seam lines. An illustration of how 3D shearlets tile the frequency domain is provided in Figure 2.

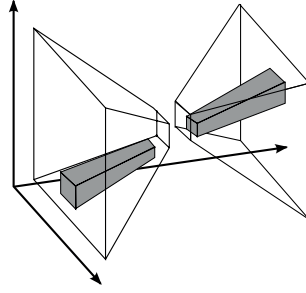


Fig. 2. Tiling of the frequency domain induced by 3D shearlets.

2.3. Previous Implementations

We now briefly review previous implementations of the shearlet transform. It should be emphasized though that all implementations so far only focussed on the parabolic case, i.e., $\alpha_j = 1$ for all scales j .

2.3.1. Fourier Domain Approaches. Band-limited shearlet systems provide a precise partition of the frequency plane due to the fact that the Fourier transforms of all elements are compactly supported and that they form a tight frame. Hence it seems most appropriate to implement the associated transforms via a *Fourier domain approach*, which aims to directly produce the same frequency tiling.

A first numerical implementation using this approach was discussed in [Easley et al. 2008] as a cascade of a subband decomposition based on the Laplacian Pyramid filter, which was then followed by a step containing directional filtering using the Pseudo-Polar Discrete Fourier Transform. Another approach was suggested in [Kutyniok et al. 2012b]. The main idea here is to employ a carefully weighted Pseudo-Polar transform with weights ensuring (almost) isometry. This step is then followed by appropriate windowing and the inverse FFT applied to each windowed part.

2.3.2. Spatial Domain Approaches. We refer to a numerical realization of a shearlet transform as a *spatial domain approach* if the filters associated with the transform are implemented by a convolution in the spatial domain. Whereas Fourier-based approaches were only utilized for band-limited shearlet transforms, the range of spatial domain-based approaches is much broader and basically justifiable for a transform based on any shearlet system. We now present the main contributions.

In the paper [Easley et al. 2008] already referred to in Subsection 2.3.1, also a spatial domain approach is discussed. This implementation utilizes directional filters, which are obtained as approximations of the inverse Fourier transforms of digitized band-limited window functions. A numerical realization specifically focussed on separable shearlet generators ψ^{sep} given by $\psi^{\text{sep}} = \psi_1 \otimes \phi_1$ – which includes certain compactly supported shearlet frames (cf. Subsection 2.2) – was derived in [Lim 2010]. This algorithm enables the application of fast transforms separably along both axes, even if the corresponding shearlet transform is not associated with a tight frame. The most faithful, efficient, and numerically stable (in the sense of closeness to tightness) digitalization of the shearlet transform was derived in [Lim 2013] by utilizing non-separable compactly supported shearlet generators, which best approximate the classical band-limited generators. In fact, the implementation of the universal shearlet transform we will discuss in this paper will be based on this work.

There exist two other approaches, which though have not been numerically tested yet. The one introduced in [Kutyniok and Sauer 2009] explores the theory of subdivision schemes inserting directionality, and leads to a version of the shearlet transform which admits an associated multiresolution analysis structure. In close relation to this algorithmic realization, in [Han et al. 2011] a general unitary extension principle is proven, which – for the shearlet setting – provides equivalent conditions for the filters to lead to a shearlet frame.

3. DIGITAL SHEARLET TRANSFORM

In this section, we will introduce and discuss the algorithms which are implemented in ShearLab 3D. We will start with a brief review of the digital wavelet transform in Subsection 3.1, which parts of the digital shearlet transform will be based upon. In Subsection 3.2 the 2D forward shearlet transform will be discussed, first the parabolic version, followed by a general version with freely chosen α . The 3D version is then presented in Subsection 3.3. Finally, the inverse shearlet transform both for 2D and 3D is detailed in Subsection 3.4.

3.1. Digital Wavelet Transform

We start by recalling the notion of the discrete Fourier transform of a sequence $\{a(n)\}_{n \in \mathbb{Z}^d} \in \ell^2(\mathbb{Z}^d)$, which is defined by

$$\hat{a}(\xi) = \sum_{n \in \mathbb{Z}^d} a(n) e^{-2\pi i n \cdot \xi}.$$

In the sequel, we will also use the notion $\bar{a}(n) = a(-n)$ for $n \in \mathbb{Z}^d$. We further wish to refer the reader not familiar with wavelets to the books [Daubechies 1992; Mallat 2008].

Let now ψ^1 and $\phi^1 \in L^2(\mathbb{R})$ be a wavelet and an associated scaling function, respectively, satisfying the two scale relations

$$\phi^1(x_1) = \sum_{n_1 \in \mathbb{Z}} h(n_1) \sqrt{2} \phi^1(2x_1 - n_1) \quad (3)$$

and

$$\psi^1(x_1) = \sum_{n_1 \in \mathbb{Z}} g(n_1) \sqrt{2} \phi^1(2x_1 - n_1). \quad (4)$$

Then, for each $j > 0$ and $n_1 \in \mathbb{Z}$, the associated wavelet function $\psi_{j,n_1}^1 \in L^2(\mathbb{R})$ is defined by

$$\psi_{j,n_1}^1(x_1) = 2^{j/2} \psi^1(2^j x_1 - n_1).$$

Now let f be a function on \mathbb{R} , for which we assume an expansion of the type

$$f^{1D}(x_1) = \sum_{n_1 \in \mathbb{Z}} f_J^{1D}(n_1) 2^{J/2} \phi_1(2^J x_1 - n_1)$$

for a fixed, sufficiently large $J > 0$. To derive a formula for the associated wavelet coefficients, for each $j > 0$, let $\{h_j(n_1)\}_{n_1 \in \mathbb{Z}}$ and $\{g_j(n_1)\}_{n_1 \in \mathbb{Z}}$ denote the Fourier coefficients of the trigonometric polynomials

$$\hat{h}_j(\xi_1) = \prod_{k=0}^{j-1} \hat{h}(2^k \xi_1) \quad \text{and} \quad \hat{g}_j(\xi_1) = \hat{g}\left(\frac{2^j \xi_1}{2}\right) \hat{h}_{j-1}(\xi_1) \quad (5)$$

with $\hat{h}_0 \equiv 1$. Then using (3) and (4) and setting $w_j \equiv g_{J-j}$ as well as $\Phi^1(n_1) = \langle \phi^1(\cdot), \phi^1(\cdot - n_1) \rangle$, the wavelet coefficients $\langle f^{1D}, \psi_{j,n_1}^1 \rangle$ can be computed by the discrete formula given by

$$\langle f^{1D}, \psi_{j,n_1}^1 \rangle = \overline{w_j} * (f_J^{1D} * \phi^1)(2^{J-j} \cdot m_1).$$

In particular, when ϕ_1 is an orthonormal scaling function, this expression reduces to

$$\langle f^{1D}, \psi_{j,n_1}^1 \rangle = (\overline{w_j} * f_J^{1D})(2^{J-j} \cdot m_1). \quad (6)$$

This last formula is the common *1D digital wavelet transform*.

This transform can now be easily extended to the multivariate case by using tensor products. Similar to the 1D case, we consider a 2D function f given by

$$f(x) = \sum_{n \in \mathbb{Z}^2} f_J(n) 2^J \phi(2^J x_1 - n_1, 2^J x_2 - n_2), \quad (7)$$

where $\phi(x) = \phi_1 \otimes \phi_1(x)$. Assume that ϕ_1 is an orthonormal scaling function. Then, using (6), it is a straightforward calculation to show that the *2D digital wavelet transform*, which computes wavelet coefficients for f , is of the form

$$\begin{aligned} \langle f, \psi_{j,m_1}^1 \otimes \phi_{j,m_1}^1 \rangle &= (\overline{g_{J-j} \otimes h_{J-j}} * f)(2^{J-j} \cdot m), \\ \langle f, \phi_{j,m_1}^1 \otimes \psi_{j,m_2}^1 \rangle &= (\overline{h_{J-j} \otimes g_{J-j}} * f)(2^{J-j} \cdot m), \end{aligned}$$

and

$$\langle f, \psi_{j,m_1}^1 \otimes \psi_{j,m_1}^1 \rangle = (\overline{g_{J-j} \otimes g_{J-j}} * f)(2^{J-j} \cdot m).$$

3.2. 2D Shearlet Transform

We will now describe the algorithmic realization of the transform associated with a universal compactly supported shearlet system $SH(\phi, \psi, \tilde{\psi}; \alpha, c)$ (as introduced in Subsection 2.2) used in ShearLab 3D. We remark that the subset $\Phi(\phi; c_1)$ is merely the scaling part coinciding with the wavelet scaling part. Moreover, it will be sufficient to consider shearlets from $\Psi(\psi; \alpha, c)$ as the same arguments apply to $\tilde{\Psi}(\tilde{\psi}; \alpha, c)$ except for switching the order of variables.

We start with the digitalization of the universal shearlet transform for the parabolic case, i.e., $\alpha_j = 1$ for all scales j , which was initially introduced in [Lim 2013]. We then extend this algorithmic approach to the situation of a general parameter $\alpha = (\alpha_j)_j$, allowing α_j to differ for each scale j .

3.2.1. The Parabolic Case. First, the shearlet generator ψ needs to be chosen. In Subsection 2.3.2, the choice of a separable shearlet generator $\psi = \psi_1 \otimes \phi_1$ was discussed, which generated a shearlet frame provided that the 1D wavelet function ψ_1 and the 1D scaling function ϕ_1 are sufficiently smooth and ψ_1 has sufficient vanishing moments. However, significantly improved numerical results can be achieved by choosing a *non-separable* generator ψ such as

$$\hat{\psi}(\xi) = P(\xi_1/2, \xi_2)\hat{\psi}^{\text{sep}}(\xi), \quad (8)$$

where the trigonometric polynomial P is a 2D fan filter (cf. [Do and Vetterli 2005], [da Cunha et al. 2006]). The reason for this is the fact, that non-separability allows the Fourier transform of ψ to have a wedge shaped essential support, thereby well approximating the Fourier transform of a classical shearlet. In particular, with a suitable choice for a 2D fan filter P , we have

$$P(\xi_1/2, \xi_2)\hat{\phi}_1(\xi_2) \approx \hat{\psi}_2\left(\frac{\xi_2}{\xi_1}\right)$$

and

$$P(\xi_1/2, \xi_2)\hat{\psi}_1(\xi_1)\hat{\phi}_1(\xi_2) \approx \hat{\psi}_1(\xi_1)\hat{\psi}_2\left(\frac{\xi_2}{\xi_1}\right)$$

where $\hat{\psi}_1(\xi_1)\hat{\psi}_2(\frac{\xi_2}{\xi_1})$ is the classical shearlet generator defined in (2). Compared with separable compactly supported shearlet generators, this property does indeed not only improve the frame bounds of the associated system, but also improves the directional selectivity significantly. Figure 3 shows some non-separable compactly supported shearlet generators both in time and frequency domain. One can in fact construct compactly supported functions ψ_1 and ϕ_1 , and a finite 2D fan filter P such that

$$\inf\{|\hat{\psi}_1(\xi_1)|^2 : 1/2 < |\xi_1| < 1\} > \delta_1, \quad \inf\{|\hat{\phi}_1(\xi_1)|^2 : -1/2 < \xi_1 < 1/2\} > \delta_2, \quad (9)$$

and

$$\inf\{|P(\xi)|^2 : 1/4 < |\xi_1| < 1/2, |\xi_2/\xi_1| < 1\} > \delta_3 \quad (10)$$

with some δ_1, δ_2 and $\delta_3 > 0$. This implies

$$|\hat{\phi}(\xi)|^2 + \sum_{j \geq 0} \sum_{|k| \leq \lceil 2^{j/2} \rceil} \left(|\hat{\psi}_{j,k,0}(\xi)|^2 + |\hat{\psi}_{j,k,0}(\xi)|^2 \right) > \min(\delta_2^2, \delta_1 \delta_2 \delta_3)$$

with $\phi = \phi_1 \otimes \phi_1$ and $\tilde{\psi}(x_1, x_2) = \psi(x_2, x_1)$. This inequality provides a lower frame bound provided that ψ_1 and ϕ_1 decay sufficiently fast in frequency and ψ_1 has sufficient vanishing moments. One can also obtain an upper frame bound with ψ generated by those 1D functions. We refer to [Kittipoom et al. 2012] for more details.

The task is now to derive a digital formulation for the computation of the associated shearlet coefficients $\langle f, \psi_{j,k,m} \rangle$ for $j = 0, \dots, J-1$ of a function f given as in (7), where

$$\psi_{j,k,m}(x) = 2^{\frac{3}{4}j} \psi(S_k A_{2^j} x - M_{c_j} m)$$

with the sampling matrix given by $M_{c_j} = \text{diag}(c_1^j, c_2^j)$. Without loss of generality, we will from now on assume that $j/2$ is integer; otherwise $\lfloor j/2 \rfloor$ would need to be taken.

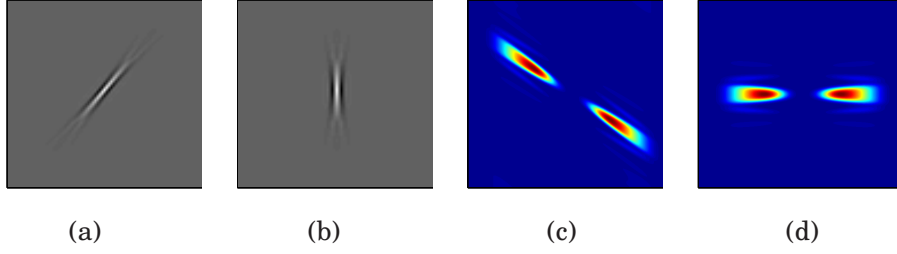


Fig. 3. Shearlets $\psi_{j,k,m}$ in the time/frequency domain. (a)–(b) : Shearlets in the spatial domain. (c)–(d) : Shearlets in the frequency domain.

We first observe that

$$S_k A_{2^j} = A_{2^j} S_{k/2^{j/2}},$$

which implies that

$$\psi_{j,k,m}(\cdot) = \psi_{j,0,m}(S_{k/2^{j/2}} \cdot) \quad (11)$$

Thus, our strategy for discretizing $\psi_{j,k,m}$ consists of the following two parts:

- (I) Faithful discretization of $\psi_{j,0,m}$ using the structure of the multiresolution analysis associated with (8).
- (II) Faithful discretization of the shear operator $S_{k/2^{j/2}}$.

We start with part (I), which will require the digital wavelet transform introduced in Subsection 3.1. Without loss of generality, we assume $M_{c_j} = Id$. The general case can be treated similarly. First, using (3), (4) and (8), we obtain

$$\begin{aligned} \hat{\psi}_{j,0,m}(\xi) &= 2^{-\frac{3}{4}j} e^{-2\pi i m \cdot A_{2^j}^{-1} \xi} \hat{\psi}(A_{2^j}^{-1} \xi) \\ &= 2^{-\frac{3}{4}j-1} e^{-2\pi i n \cdot A_{2^j}^{-1} \xi} P(A_{2^j}^{-1} Q^{-1} \xi) \hat{g}(2^{-j-1} \xi_1) \hat{h}(2^{-j/2-1} \xi_2) \hat{\phi}(A_{2^j}^{-1} 2^{-1} \xi), \end{aligned} \quad (12)$$

where $Q = \text{diag}(2, 1)$. From now on, we assume that ϕ is an orthonormal scaling function, i.e.,

$$\sum_{n \in \mathbb{Z}^2} |\hat{\phi}(\xi + n)|^2 = 1. \quad (13)$$

Applying (3) and (4) iteratively, we obtain

$$\hat{\phi}(A_{2^j}^{-1} 2^{-1} \xi) = 2^{-J+\frac{3}{4}j+1} \prod_{\ell=1}^{J-j-1} \hat{h}(2^{-j-1-\ell} \xi_1) \prod_{\ell=1}^{J-j/2-1} \hat{h}(2^{-j/2-1-\ell} \xi_2) \hat{\phi}(2^{-J} \xi).$$

Inserted in (12), it follows that

$$\hat{\psi}_{j,0,m}(\xi) = 2^{-J} e^{-2\pi i m \cdot A_{2^j}^{-1} \xi} P(A_{2^j}^{-1} Q^{-1} \xi) \hat{g}_{J-j} \otimes \hat{h}_{J-j/2}(2^{-J} \xi) \hat{\phi}(2^{-J} \xi).$$

Assuming the function f to be of the form (7), hence its Fourier transform is

$$\hat{f}(\xi) = 2^{-J} \hat{f}_J(2^{-J} \xi) \hat{\phi}(2^{-J} \xi),$$

we conclude that

$$\begin{aligned} \langle f, \psi_{j,0,m} \rangle &= 2^{-2J} \int_{\mathbb{R}^2} \hat{f}_J(2^{-J} \xi) |\hat{\phi}(2^{-J} \xi)|^2 e^{2\pi i m \cdot A_{2^j}^{-1} \xi} P^*(A_{2^j}^{-1} Q^{-1} \xi) \hat{W}_j^*(2^{-J} \xi) \hat{\phi}^*(2^{-J} \xi) d\xi \end{aligned}$$

where $W_j = g_{J-j} \otimes h_{J-j/2}$. Letting $\eta = 2^{-j}\xi$ and using (13),

$$\begin{aligned} \langle f, \psi_{j,0,m} \rangle &= \int_{\mathbb{R}^2} \hat{f}_J(\eta) |\hat{\phi}(\eta)|^2 e^{2\pi i m \cdot A_{2^j}^{-1} 2^j \eta} P^*(2^j A_{2^j}^{-1} Q^{-1} \eta) \hat{W}_j^*(\eta) d\eta \\ &= \int_{[0,1]^2} \hat{f}_J(\eta) \sum_{n \in \mathbb{Z}^2} |\hat{\phi}(\eta + n)|^2 e^{2\pi i m \cdot A_{2^j}^{-1} 2^j \eta} P^*(2^j A_{2^j}^{-1} Q^{-1} \eta) \hat{W}_j^*(\eta) d\eta \\ &= \int_{[0,1]^2} \hat{f}_J(\eta) e^{2\pi i m \cdot A_{2^j}^{-1} 2^j \eta} P^*(2^j A_{2^j}^{-1} Q^{-1} \eta) \hat{W}_j^*(\eta) d\eta. \end{aligned}$$

Thus, letting $p_j(n)$ be the Fourier coefficients of $P(2^{J-j-1}\xi_1, 2^{J-j/2}\xi_2)$ with a 2D fan filter,

$$\langle f, \psi_{j,0,m} \rangle = (f_J * (\overline{p_j * W_j}))(A_{2^j}^{-1} 2^j m). \quad (14)$$

We remark that in case $p_j \equiv 1$ this coincides with the 2D wavelet transform associated with the anisotropic scaling matrix A_{2^j} and a separable wavelet generator, while we obtain the anisotropic wavelet transform with a nonseparable wavelet generator in case p_j is a nonseparable filter. Note that in case of $M_{c_j} \neq Id$, (14) can be easily extended to obtain

$$\langle f, \psi_{j,0,m} \rangle = (f_J * (\overline{p_j * W_j}))(A_{2^j}^{-1} 2^j M_{c_j} m), \quad (15)$$

for which the sampling matrix M_{c_j} should be chosen so that $A_{2^j}^{-1} 2^j M_{c_j} m \in \mathbb{Z}^2$.

We next turn to part (II), i.e., to faithfully digitize the shear operator $S_{2^{-j/2}k}$, which will then provide an algorithm for computing $\langle f, \psi_{j,k,m} \rangle$ by using (11) combined with (14). We however face the problem that in general, the shear matrix $S_{2^{-j/2}k}$ does not preserve the regular grid \mathbb{Z}^2 , i.e.,

$$S_{2^{-j/2}k}(\mathbb{Z}^2) \neq \mathbb{Z}^2.$$

One approach to resolve this problem is to refine the regular grid \mathbb{Z}^2 along the horizontal axis x_1 by a factor of $2^{j/2}$. With this modification, the new grid $2^{-j/2}\mathbb{Z} \times \mathbb{Z}$ is now invariant under the shear operator $S_{2^{-j/2}k}$, since

$$2^{-j/2}\mathbb{Z} \times \mathbb{Z} = Q^{-j/2}(\mathbb{Z}^2) = Q^{-j/2}(S_k(\mathbb{Z}^2)) = S_{2^{-j/2}k}(2^{-j/2}\mathbb{Z} \times \mathbb{Z}).$$

Thus the operator $S_{2^{-j/2}k}$ is indeed well defined on the refined grid $2^{-j/2}\mathbb{Z} \times \mathbb{Z}$, which provides a natural discretization of $S_{2^{-j/2}k}$. This observation gives rise to the following strategy for computing sampling values of $f(S_{k2^{-j/2}} \cdot)$ for given samples $f_J \in \ell^2(\mathbb{Z}^2)$ from the function $f \in L^2(\mathbb{R}^2)$. For this, let $\uparrow 2^{j/2}$, $\downarrow 2^{j/2}$, and $*_1$ be the 1D upsampling, downsampling, and convolution operator along the horizontal axis x_1 , respectively. First, compute interpolated sample values \tilde{f}_J from $f_J(n)$ on the refined grid $2^{-j/2}\mathbb{Z} \times \mathbb{Z}$, which is invariant under $S_{k2^{-j/2}}$, by

$$\tilde{f}_J := ((f_J)_{\uparrow 2^{j/2}} *_1 h_{j/2}). \quad (16)$$

Recall that on this new grid $2^{-j/2}\mathbb{Z} \times \mathbb{Z}$, the shear operator $S_{k2^{-j/2}}$ becomes S_k with integer entries. This allows \tilde{f}_J now to be resampled by S_k , followed by reversing the previous convolution and upsampling, i.e.,

$$S_{2^{-j/2}k}^d(f_J) := \left(((\tilde{f}_J)(S_k \cdot)) *_1 \bar{h}_{j/2} \right)_{\downarrow 2^{j/2}}. \quad (17)$$

which combined with (16) performs the application of $S_{k2^{-j/2}}$ to discrete data f_J . Figure 4 illustrates how this approach effectively removes otherwise appearing aliasing effect.

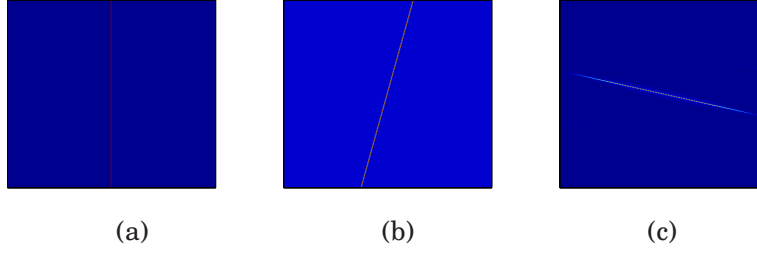


Fig. 4. (a) f_d : Digital image of $1_{\{x: x_1=0\}}$. (b) $S_{-1/4}^d(f_d)$: Sheared image. (c) Magnitude of the DFT of $S_{-1/4}^d(f_d)$.

Finally, combining (15) with the digital shear operator $S_{2^{-j/2}k}^d$ just defined by (16) and (17), yields a faithful digital shearlet transform as follows.

Definition 3.1. Let $f_J \in \ell^2(\mathbb{Z}^2)$ be the scaling coefficients given in (7). Then the *digital shearlet transform* associated with $\Psi(\psi; c)$ is defined by

$$DST_{j,k,m}^{2D}(f_J) = (\overline{\psi_{j,k}^d} * f_J)(2^J A_{2^j}^{-1} M_{c_j} m) \quad \text{for } j = 0, \dots, J-1,$$

where

$$\psi_{j,k}^d = S_{k/2^{j/2}}^d(p_j * W_j),$$

with the shearing operator defined by (16) and (17), and the sampling matrix M_{c_j} chosen so that $2^J A_{2^j}^{-1} M_{c_j} m \in \mathbb{Z}^2$.

3.2.2. The General Case. The digitalization of the shearlet transform associated with the special case of a classical cone-adapted discrete shearlet system as defined in Definition 3.1 shall now be extended to universal shearlet systems, where the parabolic scaling matrices are generalized to $A_{\alpha_j, 2^j} = \text{diag}(2^j, 2^{\alpha_j j/2})$ and $\tilde{A}_{\alpha_j, 2^j} = \text{diag}(2^{\alpha_j j/2}, 2^j)$ with $\alpha_j \in (0, 2)$ (cf. Subsection 2.2). In this general setting, the range of the shearing parameter $k \in \mathbb{Z}$ is given as $|k| \leq \lceil 2^{(2-\alpha_j)j/2} \rceil$ in each cone for each scale $j \in \mathbb{N}_0$. The associated shearlet coefficients are then given by

$$\langle f, \psi_{j,k,m} \rangle = \langle f(S_{k/2^{(2-\alpha_j)j/2}} \cdot), \psi_{j,0,m}(\cdot) \rangle, \quad (18)$$

where $\psi_{j,0,m}(\cdot) = 2^{\frac{(2+\alpha_j)j}{4}} \psi(A_{\alpha_j, 2^j} \cdot - m)$.

From now on, we retain notations from the previous section, otherwise we specify them. To digitalize (18), we slightly change the range of shearings $|k| \leq \lceil 2^{(2-\alpha_j)j/2} \rceil$ to $|k| \leq 2^{\lceil (2-\alpha_j)j/2 \rceil}$, the reason being that then the number of shearings is determined by dyadic scales. Thus, also to have an integer scaling matrix, we now consider the slightly modified version of (18) given by

$$\langle f(S_{k/2^{d_j}} \cdot), \psi_{j,0,m}(\cdot) \rangle \quad \text{with } d_j := \lceil (2 - \alpha_j)j/2 \rceil, \quad (19)$$

where

$$\psi_{j,0,m}(\cdot) = 2^{\frac{2j-d_j}{2}} \psi(A_{d_j} \cdot - m) \quad \text{and} \quad A_{d_j} := \text{diag}(2^j, 2^{j-d_j}).$$

Again we have to digitalize parts (I) and (II). Starting with part (I), the coefficients $\langle f, \psi_{j,0,m} \rangle$ can be digitalized similar to (15), except for changing A_{2^j} to A_{d_j} , i.e., changing the scaling parameter $j/2$ to $j - d_j$. Then the resulting discretization for $\psi_{j,0,m}$

essentially follows the α_j scaling operator $A_{\alpha_j, 2^j}$ in the sense that $j - d_j \approx \alpha_j(j/2)$ for sufficiently large j . In short, we obtain

$$\langle f, \psi_{j,0,m} \rangle = (f_J * (\overline{p_j} * \overline{W_j}))(A_{d_j}^{-1} 2^J M_{c_j} m), \quad (20)$$

for which we modify p_j and W_j so that $p_j(n)$ are now the Fourier coefficients of $P(2^{J-j-1}\xi_1, 2^{J-(j-d_j)}\xi_2)$ with a 2D fan filter P and $W_j = g_{J-j} \otimes h_{J-(j-d_j)}$ and the sampling matrix M_{c_j} chosen so that $A_{d_j}^{-1} 2^J M_{c_j} m \in \mathbb{Z}^2$.

Concerning part (II), we just need to slightly modify (16) and (17) to become

$$S_{k/2^{d_j}}^d(f_J) := \left(((\tilde{f}_J)(S_k \cdot)) *_1 \bar{h}_{d_j} \right)_{\downarrow 2^{d_j}} \quad (21)$$

where

$$\tilde{f}_J = ((f_J)_{\uparrow 2^{d_j}} *_1 h_{d_j}) \quad (22)$$

and filter coefficients h_{d_j} are defined as in (5).

Thus, concluding, we derive the following digitalization of the universal shearlet transform associated with the shearlets from $\Psi(\psi; \alpha, c)$.

Definition 3.2. Let $f_J \in \ell^2(\mathbb{Z}^2)$ be the scaling coefficients given in (7). Then the *digital shearlet transform* associated with $\Psi(\psi; \alpha, c)$ is defined by

$$DST_{j,k,m}^{2D}(f_J) = (\overline{\psi_{j,k}^d} * f_J)(2^J A_{d_j}^{-1} M_{c_j} m) \quad \text{for } j = 0, \dots, J-1,$$

where

$$\psi_{j,k}^d = S_{k/2^{j/2}}^d(p_j * W_j) \quad (23)$$

with the shearing operator defined by (22) and (21), p_j and W_j modified as explained above, and the sampling matrix M_{c_j} chosen so that $2^J A_{d_j}^{-1} M_{c_j} m \in \mathbb{Z}^2$.

3.3. 3D Shearlet Transform

Following the approach of the 2D case, we choose the 3D shearlet generator ψ by

$$\hat{\psi}(\xi) = \left(P(\xi_1/2, \xi_2) \hat{\psi}_1(\xi_1) \hat{\phi}_1(\xi_2) \right) \cdot \left(P(\xi_1/2, \xi_3) \hat{\phi}_1(\xi_3) \right) \quad (24)$$

so that – as it was also the idea in the 2D case – $\hat{\psi}$ well approximates the 3D band-limited shearlet generator in the sense that

$$P(\xi_1/2, \xi_2) \hat{\psi}_1(\xi_1) \hat{\phi}_1(\xi_2) \approx \hat{\psi}_1(\xi_1) \hat{\phi}_1(\xi_2/\xi_1) \quad \text{and} \quad P(\xi_1/2, \xi_3) \hat{\phi}_1(\xi_3) \approx \hat{\phi}_1(\xi_3/\xi_1).$$

Similar to the 2D case, we may choose compactly supported functions ψ_1 and ϕ_1 , and a finite 2D fan filter satisfying (9) and (10), respectively. Then we have

$$|\hat{\phi}(\xi)|^2 + \sum_{j \geq 0} \sum_{|k| \leq \lceil 2^{j/2} \rceil} \left(|\hat{\psi}_{j,k,0}(\xi)|^2 + |\hat{\tilde{\psi}}_{j,k,0}(\xi)|^2 + |\hat{\hat{\psi}}_{j,k,0}(\xi)|^2 \right) > \min(\delta_2^3, \delta_1 \delta_2^2 \delta_3^2),$$

where $|k| = \max(|k_1|, |k_2|)$, $\phi = \phi_1 \otimes \phi_1 \otimes \phi_1$, $\tilde{\psi}_{j,k,m}(x_1, x_2, x_3) = \psi_{j,k,m}(x_2, x_1, x_3)$ and $\hat{\psi}_{j,k,m}(x_1, x_2, x_3) = \psi_{j,k,m}(x_3, x_1, x_2)$. This inequality provides a lower frame bound provided that ψ_1 and ϕ_1 decay sufficiently fast in frequency and ψ_1 has sufficient vanishing moments. Also, an upper frame bound can be obtained from the fast decay rate of $\hat{\psi}$ generated by those 1D functions ψ_1 and ϕ_1 . We refer to [Kutyniok et al. 2012a] for more details.

We next discuss a digitalization of the associated shearlet coefficients $\langle f, \psi_{j,k,m} \rangle$ again only from $\Psi(\psi; \alpha, c)$, where f is given by

$$f(x) = \sum_{n \in \mathbb{Z}^3} f_J(n) 2^{J \cdot 3/2} \phi_1 \otimes \phi_1 \otimes \phi_1(2^J x - n). \quad (25)$$

3.3.1. The Parabolic Case. We start with the parabolic case, i.e., $\alpha_j = 1$ for all scales j . Recalling the 3D parabolic scaling matrix $A_{2^j} = \text{diag}(2^j, 2^{j/2}, 2^{j/2})$ from Subsection 2.2, we now consider

$$\hat{\Phi}_{j,k_1}(\xi_1, \xi_2) = (\hat{\phi}_1 \cdot P) \left(Q^{-1} \left(S_{-k_1}^T A_{2^j}^{-1}(\xi_1, \xi_2)^T \right) \right),$$

$$\hat{\Phi}_{j,k_2}(\xi_1, \xi_3) = (\hat{\phi}_1 \cdot P) \left(Q^{-1} \left(S_{-k_2}^T A_{2^j}^{-1}(\xi_1, \xi_3)^T \right) \right),$$

and

$$\gamma_{j,k,m}(\xi) = \left(\frac{m_1}{2^j} - k_1 \frac{m_2}{2^j} - k_2 \frac{m_3}{2^j} \right) \xi_1 + m_2 \frac{\xi_2}{2^{j/2}} + m_3 \frac{\xi_3}{2^{j/2}}.$$

Using the generator (24), our 3D shearlets $\psi_{j,k,m}$ are then defined by

$$\hat{\psi}_{j,k,m}(\xi) = \frac{1}{2^j} \hat{\psi}_1(\xi_1/2^j) \hat{\Phi}_{j,k_1}(\xi_1, \xi_2) \hat{\Phi}_{j,k_2}(\xi_1, \xi_3) e^{-2\pi i \gamma_{j,k,m}(\xi)}. \quad (26)$$

Since $\hat{\Phi}_{j,k_1}$ and $\hat{\Phi}_{j,k_2}$ are functions of the form of 2D shearlets, they might be discretized similar as in Definition 3.1 – though omitting the convolution with the high-pass filter $g_{J-j} -$, which gives $(*_x)$ denoting 1D convolution along the x_i axis

$$\Phi_{j,k_1}^d(n_1, n_2) = \left(S_{k_1 2^{-j/2}}^d (h_{J-j/2} *_x p_j) \right)(n_1, n_2)$$

and

$$\Phi_{j,k_2}^d(n_1, n_3) = \left(S_{k_2 2^{-j/2}}^d (h_{J-j/2} *_x p_j) \right)(n_1, n_3).$$

Finally, by (4), a 1D wavelet $2^{j/2} \psi_1(2^j \cdot)$ can be digitalized by the 1D filter g_{J-j} . This gives rise to 3D digital shearlet filters $\psi_{j,k}^d$ specified in the definition below, which discretize $\psi_{j,k,m}$ from (26). Summarizing, our digitalization of the shearlet transform associated with the shearlets from $\Psi(\psi; c)$ (i.e., the parabolic case) is defined as follows.

Definition 3.3. Let $f_J \in \ell^2(\mathbb{Z}^3)$ be the scaling coefficients given in (25), and retain the definitions and notions of this subsection. Then the *digital shearlet transform* associated with $\Psi(\psi; c)$ is defined by

$$DST_{j,k}^{3D}(f_J)(m) = (f_J * \overline{\psi_{j,k}^d})(\tilde{m}) \quad \text{for } j = 0, \dots, J-1,$$

where the 3D digital shearlet filters $\psi_{j,k}^d$ are defined by

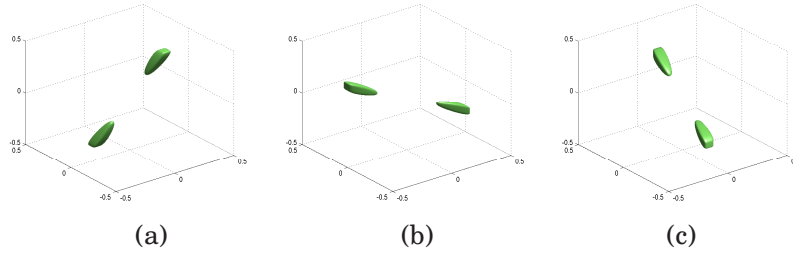
$$\hat{\psi}_{j,k}^d(\xi) = \hat{g}_{J-j}(\xi_1) \hat{\Phi}_{j,k_1}^d(\xi_1, \xi_2) \hat{\Phi}_{j,k_2}^d(\xi_1, \xi_3)$$

and

$$\tilde{m} = (2^{J-j} c_1^j m_1, 2^{J-j/2} c_2^j m_2, 2^{J-j/2} c_3^j m_3)$$

with the sampling constants c_1^j, c_2^j and c_3^j chosen so that $\tilde{m} \in \mathbb{Z}^3$.

The chosen 3D digital shearlet filters are illustrated in Figure 5.


 Fig. 5. 3D digital shearlets in the frequency domain: $\hat{\psi}_{j,k}^d$

3.3.2. The General Case. We now describe the digitalization of the 3D universal shearlet transform. Similar to the 2D situation, we first slightly modify the 3D scaling matrix $A_{\alpha_j, 2^j}$ to consider integer matrices $A_{d_j} = \text{diag}(2^j, 2^{j-d_j}, 2^{j-d_j})$ with $d_j = \lceil (2 - \alpha_j)j/2 \rceil$ for each scale $j \in \mathbb{N}_0$. Also the shearing parameters $k_1, k_2 \in \mathbb{Z}$ range from $-2^{\lceil (2-\alpha_j)j/2 \rceil}$ to $2^{\lceil (2-\alpha_j)j/2 \rceil}$. Again following the 2D approach, the 3D digital shearlet filters $\psi_{j,k}^d$ defined in Definition 3.3 are generalized by modifying Φ_{j,k_1}^d and Φ_{j,k_2}^d to be

$$\Phi_{j,k_1}^d(n_1, n_2) = \left(S_{k_1 2^{-d_j}}^d (h_{J-(j-d_j)} * p_j) \right)(n_1, n_2) \quad (27)$$

and

$$\Phi_{j,k_2}^d(n_1, n_3) = \left(S_{k_2 2^{-d_j}}^d (h_{J-(j-d_j)} * p_j) \right)(n_1, n_3), \quad (28)$$

where $p_j(n)$ are the Fourier coefficients of $P(2^{J-j-1}\xi_1, 2^{J-(j-d_j)}\xi_2)$ with a 2D fan filter P , the 1D filter $h_{J-(j-d_j)}$ is defined as in (5), and $S_{k 2^{-d_j}}^d$ is the discrete shear operator defined in (21). Thus, the 3D digital shearlet transform associated with universal shearlets is defined as described in the following definition.

Definition 3.4. Let $f_J \in \ell^2(\mathbb{Z}^3)$ be the scaling coefficients given in (25), and retain the definitions and notions of this subsection. Then the *digital shearlet transform* associated with $\Psi(\psi; \alpha, c)$ is defined by

$$DST_{j,k}^{3D}(f_J)(m) = (f_J * \overline{\psi}_{j,k}^d)(\tilde{m}) \quad \text{for } j = 0, \dots, J-1,$$

where the 3D digital shearlet filters $\psi_{j,k}^d$ are defined using (27) and (28) by

$$\hat{\psi}_{j,k}^d(\xi) = \hat{g}_{J-j}(\xi_1) \hat{\Phi}_{j,k_1}^d(\xi_1, \xi_2) \hat{\Phi}_{j,k_2}^d(\xi_1, \xi_3)$$

and

$$\tilde{m} = (2^{J-j} c_1^j m_1, 2^{J-j/2} c_2^j m_2, 2^{J-j/2} c_3^j m_3)$$

with the sampling constants c_1^j, c_2^j and c_3^j chosen so that $\tilde{m} \in \mathbb{Z}^3$.

3.4. Inverse Shearlet Transform

In this section, we define an inverse digital shearlet transform, which provides a stable reconstruction of $f_J \in \ell^2(\mathbb{Z}^d)$ from the shearlet coefficients obtained by the digital shearlet transforms from Subsections 3.2 and 3.3. For this, we will consider only the 2D parabolic case retaining notations of Subsection 3.2.1, since this can be extended to the general 2D case as well as the 3D case in a straightforward manner.

We first observe that in general, the forward shearlet transform defined in Definition 3.2 can be inverted by a frame reconstruction algorithm based on the conjugate gradient method due to frame property of shearlets – see [Mallat 2008].

It seems impossible to obtain a direct reconstruction formula unless we skip subsampling, which would then lead to a highly redundant transform. However, one possibility was indeed recently discovered in [Kutyniok and Lim 2013] in the parabolic situation, which we now describe. In this approach, we first set $c_1^j = 2^{j-J}$ and $c_2^j = 2^{j/2-J}$ in Definition 3.1. In this situation, the digital shearlet transform is merely a 2D convolution with shearlet filters, yielding a shift-invariant linear transform. Hence, for $f_J \in \ell^2(\mathbb{Z}^2)$, the digital shearlet transform takes the form

$$DST_{j,k,m}^{2D}(f_J) = f_J * \overline{\psi}_{j,k}^d(m). \quad (29)$$

As indicated before, the digital shearlet filters $\tilde{\psi}_{j,k}^d$ corresponding to $\tilde{\psi}_{j,k,m}$ are derived by switching the order of variables, which implies that the same convolution formula as (29) holds for the shearlet transform associated with $\tilde{\Psi}(\tilde{\psi}; c)$, and we can define

$$\widetilde{DST}_{j,k,m}^{2D}(f_J) = f_J * \overline{\tilde{\psi}}_{j,k}^d(m). \quad (30)$$

Let us now select separable low-pass filter by

$$\hat{\phi}^d(\xi) = \hat{h}_J(\xi_1) \cdot \hat{h}_J(\xi_2),$$

set

$$\hat{\Psi}^d(\xi) := |\hat{\phi}^d(\xi)|^2 + \sum_{j=0}^{J-1} \sum_{|k| \leq 2^{\lceil j/2 \rceil}} \left(|\hat{\psi}_{j,k}^d(\xi)|^2 + |\hat{\tilde{\psi}}_{j,k}^d(\xi)|^2 \right), \quad (31)$$

and also define dual shearlet filters by

$$\hat{\varphi}^d(\xi) = \frac{\hat{\phi}^d(\xi)}{\hat{\Psi}^d(\xi)}, \quad \hat{\gamma}_{j,k}^d(\xi) = \frac{\hat{\psi}_{j,k}^d(\xi)}{\hat{\Psi}^d(\xi)}, \quad \hat{\tilde{\gamma}}_{j,k}^d(\xi) = \frac{\hat{\tilde{\psi}}_{j,k}^d(\xi)}{\hat{\Psi}^d(\xi)}. \quad (32)$$

Using (29) and (30), the Fourier transform of f_J can be written as

$$\begin{aligned} \hat{f}_J &= \frac{\hat{f}_J}{\hat{\Psi}^d} \left(|\hat{\phi}^d|^2 + \sum_{j=0}^{J-1} \sum_{|k| \leq 2^{\lceil j/2 \rceil}} \left(|\hat{\psi}_{j,k}^d|^2 + |\hat{\tilde{\psi}}_{j,k}^d|^2 \right) \right) \\ &= \hat{f}_J \cdot (\hat{\phi}^d)^* \frac{\hat{\phi}^d}{\hat{\Psi}^d} + \sum_{j=0}^{J-1} \sum_{|k| \leq 2^{\lceil j/2 \rceil}} \left(\hat{f}_J \cdot (\hat{\psi}_{j,k}^d)^* \frac{\hat{\psi}_{j,k}^d}{\hat{\Psi}^d} + \hat{f}_J \cdot (\hat{\tilde{\psi}}_{j,k}^d)^* \frac{\hat{\tilde{\psi}}_{j,k}^d}{\hat{\Psi}^d} \right) \\ &= \hat{f}_J \cdot (\hat{\phi}^d)^* \hat{\varphi}^d + \sum_{j=0}^{J-1} \sum_{|k| \leq 2^{\lceil j/2 \rceil}} \mathcal{F}(DST_{j,k,m}^{2D}(f_J)) \hat{\gamma}_{j,k}^d + \mathcal{F}(\widetilde{DST}_{j,k,m}^{2D}(f_J)) \hat{\tilde{\gamma}}_{j,k}^d, \end{aligned}$$

where $\mathcal{F} : \ell^2(\mathbb{Z}^2) \rightarrow L^2([0, 1]^2)$ is defined as the (discrete time) Fourier transform and $*$ as a superscript denotes the complex conjugate. These considerations then yield the reconstruction formula given by

$$\begin{aligned} f_J &= (f_J * \overline{\phi}^d) * \varphi^d + \sum_{j=0}^{J-1} \sum_{|k| \leq 2^{\lceil j/2 \rceil}} (DST_{j,k,m}^{2D}(f_J)) * \gamma_{j,k}^d \\ &\quad + \sum_{j=0}^{J-1} \sum_{|k| \leq 2^{\lceil j/2 \rceil}} (\widetilde{DST}_{j,k,m}^{2D}(f_J)) * \tilde{\gamma}_{j,k}^d. \end{aligned} \quad (33)$$

Now turning to the more general situation of universal shearlet systems, it can be easily observed that the definition of the dual shearlet filters from (32) can be extended using the generalized digital shearlet filters $\psi_{j,k}^d$ from Definition 3.2. In addition, the definition of $\hat{\Psi}^d$ in (31) then needs to be generalized to

$$\hat{\Psi}^d(\xi) =: |\hat{\phi}^d(\xi)|^2 + \sum_{j=0}^{J-1} \sum_{|k| \leq 2^{d_j}} \left(|\hat{\psi}_{j,k}^d(\xi)|^2 + |\hat{\tilde{\psi}}_{j,k}^d(\xi)|^2 \right)$$

with the generalized digital shearlet filters $\psi_{j,k}^d$ in this case.

4. IMPLEMENTATION: SHEARLAB 3D

An implementation of the digital transforms

- 2D Digital Shearlet Transform,
- 3D Digital Shearlet Transform,
- Forward 2D Digital Shearlet Transform,
- Inverse 3D Digital Shearlet Transform,

described in Section 3 is provided in the MATLAB toolbox ShearLab 3D, which can be downloaded from www.shearlab.org. ShearLab 3D requires the Signal Processing Toolbox and the Image Processing Toolbox of MATLAB. If additionally the Parallel Computing Toolbox is available, CUDA-compatible NVidia graphics cards can be used to gain a significant speed up.

The ShearLab 3D toolbox provides codes to compute the digital shearlet transform of arbitrarily sized two- and three-dimensional signals according to the formulas in Definitions 3.2 and 3.3 as well as the inverse shearlet transform (33). Applying the convolution theorem, these formulas can be computed by multiplying conjugated digital shearlet filters $\overline{\psi_{j,k}^d}$, their duals $\gamma_{j,k}^d$, and the given signal f_J in the frequency domain.

We now provide more details on the forward (Subsection 4.1) and inverse transform (Subsection 4.2), provide a brief example for a potential use case (Subsection 4.3), discuss the possibility to avoid numerical instabilities along the seam lines (Subsection 4.4), and compute the complexity of our algorithms in Subsection 4.5.

4.1. Forward Transform

A schematic descriptions of the forward transform is given in Algorithm 1.

As can be seen from Algorithm 1, the computation of a shearlet decomposition of a 2D signal $f \in \ell_2(\mathbb{Z}^2)$ with ShearLab 3D requires the following input parameters:

- *nScales*: The number of scales of the shearlet system associated with the desired decomposition. Each scale corresponds to a ring-like passband in the frequency plane that is constructed from the quadrature mirror filter pair defined via the lowpass filter *quadratureMirrorFilter*. These frequency bands can then be further partitioned into directionally sensitive elements using a 2D directional filter. Note that *nScales* can also be viewed as the upper bound of the parameter j in Definitions 3.2 and 3.3. Naturally, increasing the number of scales significantly increases the redundancy of the corresponding shearlet system.
- *shearLevels*: A vector of size *nScales*, specifying for each scale the fineness of the partitioning of the corresponding ring-like passband. The larger the shear level at a specific scale, the more differently sheared atoms will live on this scale with increasingly smaller essential support sizes in the frequency domain. To be precise, let d_j be the j -th component in *shearLevels*. Then, in the 2D case this choice will generate the shearlet filters $\psi_{j,k}^d$ defined in Definition 3.2 and associated with the

ALGORITHM 1: ShearLab 3D forward transform

Input: A signal $f \in \mathbb{R}^{X \times Y \times Z}$, the number of scales $nScales \in \mathbb{N}$, a vector $shearLevels \in \mathbb{N}^{nScales}$ specifying the number of differently sheared filters on each scale, a matrix $directionalFilter$ specifying the directional filter P (compare equation (23)) and a vector $quadratureMirrorFilter$ describing the lowpass filter h_1 of a quadrature mirror filter pair.

Output: Coefficients $shearletCoeffs \in \mathbb{R}^{X \times Y \times Z \times R}$ where R denotes the redundancy of the applied shearlet system.

```
// Compute shearlet filters in the frequency domain of size  $X \times Y \times Z$  according to
// the parameters  $nScales$  and  $shearLevels$ .
shearletFilters :=
computeShearletFilters( $X, Y, Z, nScales, shearLevels, directionalFilters, quadratureMirrorFilter$ );

// Compute frequency representation of the input signal.
 $f_{freq} := \text{FFT}(f)$ ;

// For each shearlet filter, compute a three-dimensional vector (or two-dimensional,
// for 2D input) of shearlet coefficients of size  $X \times Y \times Z$  as the convolution of
// the time-domain representation of the shearlet filter with the input signal  $f$ .
// According to the convolution theorem, this can be done by pointwise
// multiplication (denoted by  $.*$ ) in the frequency domain.
for  $i := 1$  to  $R$  do
    |  $shearletCoeffs(i) := \text{IFFT}(shearletFilters(i) .* \overline{f_{freq}})$ ;
end
```

scaling matrix A_{d_j} . With this choice, the range of shearing parameters k is given by $|k| \leq 2^{d_j}$ for each cone, which generates $2(2 \cdot 2^{d_j} + 1)$ shearlet filters for each scale j .

- **directionalFilter:** A 2D directional filter that is used to partition the passbands of the several scales and hence serves as the basis of the directional 'component' of the shearlets. Our default choice in ShearLab 3D is a maximally flat 2D fan filter (see [da Cunha et al. 2006] and Figure 9). Other directional filters can for instance be constructed using the *dfilters* method from the Nonsubsampled Contourlet Toolbox. This value corresponds to the trigonometric polynomial P from equation (8).
- **quadratureMirrorFilter:** A 1D lowpass filter defining a quadrature mirror filter pair with the corresponding highpass filter and thereby a wavelet multiresolution analysis. These filters induce the passbands associated with the several scales, the number of which is defined by the parameter $nScales$. The default choice in ShearLab 3D is a symmetric maximally flat 9-tap lowpass filter (for an extensive discussion of the properties of this filter, see Subsection 5.1); but basically, any lowpass filter can be used here. This parameter corresponds to h_1 in equation (5).

Given the parameters $nScales$, $shearLevels$, $quadratureMirrorFilter$ and $directionalFilter$, ShearLab 3D can compute a set of 2D digital shearlet filters whose inner products with a given 2D signal (and all its translates) are the desired shearlet coefficients (see Algorithm 1). As each shearlet coefficient corresponds to one shearlet with a specific scale, a specific shearing, and a specific translation, the total number of coefficients computed by one shearlet decomposition is $X \cdot Y \cdot R$, where X and Y denote the size of the given signal (and therefore the number of different translates) and R denotes the redundancy of the shearlet system, which is defined by the parameters $nScales$ and $shearLevels$. In fact, the redundancy R including the low

frequency part is given by

$$R = 1 + \sum_{j=j_0}^{j_0+nScales-1} 2(2 \cdot 2^{d_j} + 1),$$

where j_0 is the coarsest scale $j = j_0$ for the shearlet transform, and one can specify any nonnegative integer for j_0 .

Let us now consider the 3D situation. Due to the formula for $\psi_{j,k}^d$ in Definition 3.3, we know that a 3D digital shearlet filter can be constructed by combining two 2D digital shearlet filters living on the same scale but with possibly differing shearings. Therefore, the input parameters for a three-dimensional decomposition are the same as in the 2D case but their meanings slightly differ. Where in 2D, each scale corresponds to a ring-like passband, in the 3D case each scale is associated with a sphere-like passband in the 3D frequency domain with the parameter $nScales$ defining the number of such spheres. The parameter $shearLevels$ on the other hand still defines the number of differently sheared atoms on one scale but there are two shearing parameters $k = (k_1, k_2)$ for a shearlet in 3D and the 3D frequency domain is partitioned in three pyramids instead of two cones.

To be more precise, let d_j be a j -th component in $shearLevels$. Then the shearlet filters $\psi_{j,k}^d$ defined in Definition 3.3 and associated with the 3D scaling matrix A_{d_j} are generated. In this case, the range of shearing parameters $k = (k_1, k_2)$ is given by $\max\{|k_1|, |k_2|\} \leq 2^{d_j}$ for each pyramid, which gives $3(2 \cdot 2^{d_j} + 1)^2$ shearlet filters for each scale j . Thus, in the 3D case, the redundancy R is given by

$$R = 1 + \sum_{j=j_0}^{j_0+nScales-1} 3(2 \cdot 2^{d_j} + 1)^2.$$

4.2. Inverse Transform

A schematic description of the inverse transform is given in Algorithm 2.

ALGORITHM 2: ShearLab 3D inverse transform

Input: A set of shearlet coefficients $shearletCoeffs \in \mathbb{R}^{X \times Y \times Z \times R}$ and dual shearlet filters $dualFilters \in \mathbb{C}^{X \times Y \times Z \times R}$ in the frequency domain, where R denotes the redundancy of the corresponding shearlet system.

Output: A reconstructed signal $f_{rec} \in \mathbb{R}^{X \times Y \times Z}$.

$f_{rec} := 0$

// The reconstructed signal f_{rec} is computed as the sum over all convolutions of the dual shearlet filters with their corresponding three-dimensional vectors (two-dimensional for 2D data) of shearlet coefficients. Due to the convolution theorem, this can be computed using pointwise multiplication (denoted by $.*$) in the frequency domain.

for $i := 1$ **to** R **do**

$f_{rec} := f_{rec} + \text{FFT}(shearletCoeffs(i)).*dualFilters(i);$

end

$f_{rec} := \text{IFFT}(f_{rec});$

To perform an inverse transform in both 2D and 3D (see algorithm 2), ShearLab 3D requires a set of shearlet coefficients and the corresponding digital shearlet filters from which the dual filters can be computed according to formula (32). The reconstructed signal is then the sum over all dual filters multiplied with their corresponding coefficients.

4.3. Example

We next provide a brief example of how to use ShearLab 3D to decompose and reconstruct a 2D signal in MATLAB. Figure 6 shows the MATLAB coding side, and Figure 7 the visual outcome.

```
nScales = 4;
shearLevels = [1,1,2,2];
% this flag is used to determine whether CUDA is used or not.
useGPU = 0;
% this flag can be used to inhibit the omission of shearlet filters on the seam lines.
% it is optional with default value 0.
fullSystem = 0;

% load the lenna image.
img = double(imread('lenna.jpg'));

% construct digital shearlet filters with a default 1D lowpass and
% a default 2D directional filter.
system = SLgetShearletSystem2D(useGPU, size(img,1), size(img,2), nScales, shearLevels, fullSystem);

% compute shearlet decomposition
shearletCoefficients = SLsheardec2D(img, system);

% compute shearlet reconstruction
reconstruction = SLshearrec2D(shearletCoefficients, system);
```

Fig. 6. Decomposition and reconstruction of a 2D image in MATLAB using ShearLab 3D. The applied shearlet system has four scales and the array *shearLevels* induces a parabolic scaling. The array *shearletCoefficients* is three-dimensional in the 2D and four-dimensional in the 3D case. In both cases, the last dimension enumerates all digital shearlet filters within the specified system with different shearing parameters k and scaling parameters j while the first two or three dimensions are associated with the translates of one single shearlet (see Figure 7).

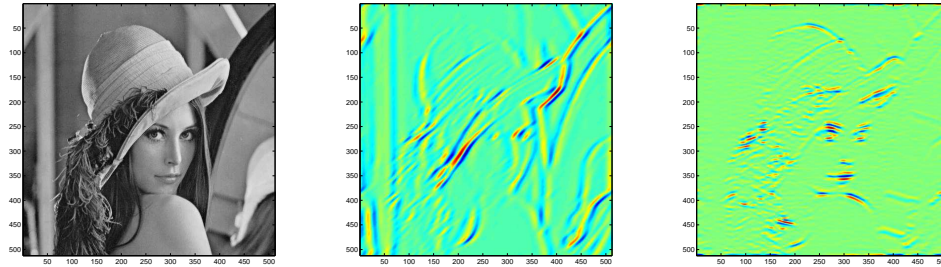


Fig. 7. The two images to the right show all shearlet coefficients of the translates of two different shearlets. The used system has four scales, a redundancy of 49 and was specified with $nScales = 4$ and $shearLevels = [1, 1, 2, 2]$. The shearlet corresponding to the coefficients in the centered picture has a scale parameter $j = 1$ a shearing parameter $k = 2$ and lives on the horizontal frequency cones. The shearlet corresponding to the coefficients plotted in the rightmost image has a scale parameter $j = 2$, a shearing parameter $k = 0$ and lives on the vertical frequency cones.

4.4. Omission of Boundary Shearlets

Let now $\psi_{j,k}^d$ and $\tilde{\psi}_{j,k}^d$, $k = \pm d_j$, be the four shearlet filters as defined in Definition 3.2. We notice that the support of each of those filters is concentrated on the boundary of the horizontal cones (or the vertical cones). Even more, the filter $\psi_{j,k}^d$ is almost identical to $\tilde{\psi}_{j,k}^d$ when $k = \pm d_j$, which is illustrated in Figure 8. For this reason, in ShearLab 3D

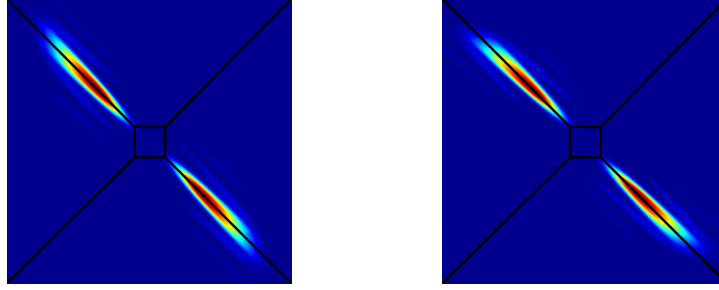


Fig. 8. The magnitude frequency response of the maximally sheared shearlet in the vertical cones (right image) is almost equal to the response of the corresponding shearlet in the horizontal cones (left image). In most cases, one of these filters can be omitted to decrease the redundancy of a shearlet system.

the boundary shearlet filters in the vertical cones are removed for each scale j in order to improve stability as well as efficiency. This leads to $2(2 \cdot 2^{d_j} + 1) - 2 = 2^{d_j+2}$ shearlet filters at each scale j . As an example, letting $d_j = \lceil j/2 \rceil$ yields 8, 8, 16, 16 shearlet filters for $j = 1, 2, 3, 4$ (see Figure 10), which corresponds to the parabolic case. In this example, the redundancy R can be computed to be $1 + 8 + 8 + 16 + 16 = 49$.

A similar strategy can be applied in the 3D case: All boundary shearlet filters whose frequency support is concentrated on the boundary of two pyramids among three are removed, yielding

$$3(2 \cdot 2^{d_j} + 1)^2 - 6(2 \cdot 2^{d_j} + 1) + 4$$

3D shearlet filters for each scale j . Again as an example, consider $d_j = \lceil j/2 \rceil$. In this case, we have 49, 49, 193 shearlet filters for $j = 1, 2, 3$ (compare also Figure 10), which corresponds to the parabolic case.

It should be mentioned that in ShearLab 3D the user has the option of including those boundary elements again if needed.

4.5. Complexity

Since convolution can be used, both the decomposition and the reconstruction algorithm reduce to multiple computations of the Fast Fourier Transform. Thus, their complexity is given by $\mathcal{O}(R \cdot N \log(N))$, where $R \in \mathbb{N}$ is the redundancy of the specific digital shearlet system, i.e., the number of digital shearlet filters $\psi_{j,k}^d$.

5. NUMERICAL EXPERIMENTS

This section is devoted to an extensive set of numerical experiments. The parameters in ShearLab 3D chosen for those results are specified in Subsection 5.1, followed by a detailed description of the transforms we compare our results to (see Subsection 5.2). We then focus on the following problems: 2D/3D denoising, 2D/3D inpainting, and also 2D decomposition of point and curvelike structures, which are the contents of Subsections 5.3 to 5.7. The results of the numerical experiments are discussed in Subsection 5.8.

All experiments have been performed with MATLAB 2013a on an Intel(R) Core(TM)2 Duo CPU E6750 processor with 2.66GHz and a NVIDIA GeForce GTX 650 Ti graphics card with 2GB RAM. The scripts and input data for all experiments are available at www.shearlab.org in support of the idea of reproducible research.

5.1. Selection of Parameters

As discussed in Subsection 3.2, the construction of a 2D digital shearlet filter $\psi_{j,k}^d$, requires a 1D lowpass filter h_1 and a 2D directional filter P , compare equation (23). The 1D filter h_1 defines a wavelet multiresolution analysis – and thereby the highpass filter g_1 –, whereas the trigonometric polynomial P is used to ensure a wedge shape of the essential frequency support of $\psi_{j,k}^d$. The choice of these filters certainly significantly impacts crucial properties of the generated digital shearlet system such as frame bounds and directional selectivity.

Our choice for h_1 – from now on denoted by $h_{ShearLab}$ – is a maximally flat, i.e. a maximum number of derivatives of the magnitude frequency response at 0 and π vanish, and symmetric 9-tap lowpass filter¹ which is normalized such that $\sum_n h_{ShearLab}(n) = 1$. For an illustration, we refer to Figure 9(a) and (b). This filter has two vanishing moments, i.e. $\int \phi(x)x^k = 0$ for $k \in \{0, 1\}$. While there is no symmetric, compactly supported, and orthogonal wavelet besides the Haar wavelet, the renormalized filter $\sqrt{2}h_{ShearLab}$ at least approximately fulfills the orthonormality condition, which is

$$\left| 2 \sum_n h_{ShearLab}(n) h_{ShearLab}(n + 2l) - \delta_{l0} \right| \leq 0.0018 \quad \text{for all } l \in \mathbb{Z},$$

with δ denoting Kronecker's delta. We remark that by choosing $h_{ShearLab}$ to be maximally flat, the amount of ripples in the digital filter $\psi_{j,k}^d$ is significantly reduced. This leads to an improved localization of the associated digital shearlets in the frequency domain. The highpass filter g_1 , hereafter denoted by $g_{ShearLab}$, is certainly chosen to be the associated mirror filter, that is

$$g_{ShearLab}(n) = (-1)^n \cdot h_{ShearLab}(n).$$

We would like to mention that the filter coefficients $h_{ShearLab}$ are quite similar to those of the Cohen-Daubechies-Feauveau (CDF) 9/7 wavelet [Cohen et al. 1992], which is used in the JPEG 2000 standard. While the CDF 9/7 wavelet has four vanishing moments and higher degrees of regularity both in the Hölder and Sobolev sense, trading these advantageous properties for maximal flatness seems to be the optimal choice for most applications.

For the trigonometric polynomial $P_{ShearLab}$, we use the maximally flat 2D fan filter² described in [da Cunha et al. 2006]. This filter is illustrated in Figure 9(c).

For the numerical experiments, we used two different digital shearlet systems in both the 2D and 3D case, the reason being that this allows us to demonstrate how different degrees of redundancy influence the performance of ShearLab 3D. The two considered 2D systems named $SL2D_1$ and $SL2D_2$, similarly $SL3D_1$ and $SL3D_2$ in 3D, are constructed as follows. Notice that they all correspond to the case $\alpha = 1$, hence the parabolic case.

The system $SL2D_1$ has four scales with four differently oriented digital shearlet filters on scales one and two, and eight directions in each of the higher scales. Including the 2D lowpass filter, the total redundancy of this system is 25. We remark that a maximally sheared filter within the horizontal cones has always an almost identical counterpart contained in the vertical cones that can be omitted without affecting the performance in most applications. The system $SL2D_2$ also consists of four scales, but

¹The MATLAB command `design(fdesign.lowpass('N,F3dB',8,0.5),'maxflat')` generates the 9-tap filter $h_{ShearLab}$, whose approximate values are $h_{ShearLab} = (0.01049, -0.02635, -0.05178, 0.27635, 0.58257, \dots)$.

²The 2D fan filter $P_{ShearLab}$ can be obtained in MATLAB using the Nonsubsampled Contourlet Toolbox by the statement `fftshift(fft2(modulate2(dfilters('dmaxflat4','d')./sqrt(2),'c')))`.

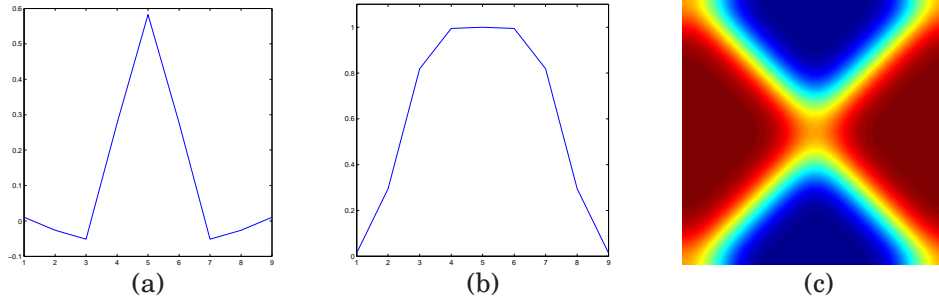


Fig. 9. (a) The coefficients of the 1D lowpass filter h_{ShearLab} . (b) Magnitude frequency response of h_{ShearLab} . (c) Magnitude response of the 2D fan filter P_{ShearLab} .

in contrast to $SL2D_2$ has a redundancy of 49 with 8, 8, 16 and 16 differently oriented shearlets on the respective scales.

In the 3D experiments, we used the three-scale digital systems $SL3D_1$ with 13, 13 and 49 directions on scales one, two and three as well as the system $SL3D_2$ with 49, 49 and 193 differently oriented shearlet filters on the corresponding scales. The total redundancy of $SL3D_1$ is 76, while $SL3D_2$ contains 292 different digital 3D shearlet filters. These numbers along with other properties of these systems are compiled in Figure 10.

| | Scales | Directions | Redundancy | A | B | B/A |
|----------|--------|-------------|------------|--------|--------|--------|
| $SL2D_1$ | 4 | (4,4,8,8) | 25 | 0.0893 | 1.0000 | 11.19 |
| $SL2D_2$ | 4 | (8,8,16,16) | 49 | 0.0669 | 1.0000 | 14.94 |
| $SL3D_1$ | 3 | (13,13,49) | 76 | 0.0075 | 1.0000 | 133.39 |
| $SL3D_2$ | 3 | (49,49,193) | 292 | 0.0045 | 1.0000 | 220.84 |

Fig. 10. Properties of the digital shearlet systems $SL2D_1$, $SL2D_2$, $SL3D_1$, and $SL3D_2$ used in our numerical experiments. Columns A and B show approximates of the lower and upper frame bounds.

5.2. Systems for Comparison

In a total of five different experiments, we compare the transforms associated with the shearlet systems $SL2D_1$, $SL2D_2$, $SL3D_1$, and $SL3D_2$ to various transforms similarly associated with a specific representation system. In each of these experiments, an algorithm based on a sparse representation of the input data is used to complete a certain task like image denoising or image inpainting. In order to get a meaningful comparison, we simply run the same algorithm with the same input several times for each of the transforms – i.e., with the associated representation system – for computing the sparse representation at each execution. To assess the performance of a sparse representation scheme, for each of the tasks we introduce a performance measure for the quality of the output and also measure the overall running time of the algorithm.

The transforms considered in our experiments besides the digital shearlet transform implemented in ShearLab 3D are:

- *Nonsubsampled Shearlet Transform (NSST, 2D & 3D).*

The NSST was introduced in 2006 by Labate, Easley, and Lim [Easley et al. 2008] and later extended to a 3D transform by Negi and Labate [Negi and Labate 2012]. It is also based on the theory of shearlets and uses the nonsubsampled Laplacian pyramid transform with specially designed bandpass filters [da Cunha et al. 2006] to decompose input data into several high-frequency layers and a low-frequency

part, while directional filters are constructed on the pseudo-polar grid from a certain window function, e.g. the Meyer wavelet window. The main conceptional difference to ShearLab 3D is that these directional filters are not compactly supported in the time domain. Still, by representing the directional filters with small matrices, the NSSST also manages to construct digital shearlet filters that are highly localized in the time domain. An implementation is publicly available at <http://www.math.uh.edu/~dlabate/software.html>.

- *Nonsubsampled Contourlet Transform (NSCT, 2D).*

The NSCT was developed by da Cunha, Zhou, and Do [da Cunha et al. 2006] and uses a nonsubsampled pyramid decomposition as well as a directional filter bank based on two-channel fan filter banks to construct directionally sensitive digital filters on several scales. It was shown that the NSCT can be applied to construct frames for $\ell_2(\mathbb{Z}^2)$ and that (band-limited) contourlets can achieve the optimal approximation rate for cartoon-like images [Do and Vetterli 2005]. The Nonsubsampled Contourlet Toolbox can be downloaded from <http://www.mathworks.com/matlabcentral/fileexchange/10049-nonsubsampled-contourlet-toolbox>.

- *Fast Discrete Curvelet Transform (FDCT, 2D).*

Curvelets were first introduced in 1999 by Candès and Donoho [Candès and Donoho 1999a] with the goal of constructing a non-adaptive frame of representing functions providing optimal approximation rates for cartoon-like images. Indeed, the curvelet transform was the first non-adaptive method published to achieve this and can be viewed as a precursor to the theory of shearlets. The most significant conceptional differences are that the shearlet transform is associated with a single (or finite set of) generating function(s) that can be subject to anisotropic scaling and shearing, while curvelet atoms are constructed by rotating 'mother' curvelets that exist on each scale. The FDCT used in our experiments is described in [Candès et al. 2006] and can be downloaded from <http://www.curvelet.org/software.html>.

- *Surfacelet Transform (SURF, 3D).*

In the surfacelet transform, first published by Do and Lu in 2007 [Do and Lu 2007], the two-dimensional Bamberger and Smith directional filter bank [Bamberger and Smith 1992] (which is also used in the NSCT) is extended to higher dimensions. Together with a pyramid transform similar to one developed by Simoncelli and Freeman [Simoncelli et al. 1992], a directional selective three-dimensional multiscale transform can be constructed. An implementation is available at <http://www.mathworks.com/matlabcentral/fileexchange/14485-surfacelet-toolbox>.

- *Stationary Wavelet Transform (SWT, 2D).*

The SWT, also known as *algorithme à trous*, is a redundant and translation invariant version of the discrete wavelet transform. Instead of dyadically downsampling the signal at each transition from one scale to another, the filter coefficients are dyadically upsampled. In our experiments, we used the method SWT2 from the MATLAB Wavelet Toolbox.

5.3. Image Denoising

As input, we consider several grayscale images of size 512x512 that are distorted with Gaussian white noise. In order to denoise these images, we use hard thresholding on the coefficients of a sparse representation scheme before computing the reconstruction.

That is, for an image $f \in \ell^2(\mathbb{Z}^2)$ and

$$f_{noisy}(i, j) = f(i, j) + e(i, j)$$

where $e(i, j) \sim \mathcal{N}(0, \sigma^2)$, we compute

$$f_{denoised} = \mathcal{T}^{-1} T_\delta \mathcal{T} f_{noisy},$$

where \mathcal{T} is the forward and \mathcal{T}^{-1} is the inverse transform associated with a certain sparse representation scheme and T_δ is the hard thresholding operator given by

$$(T_\delta x)(n) = \begin{cases} x(n) & \text{if } |x(n)| \geq \delta, \\ 0 & \text{else.} \end{cases} \quad (34)$$

In order to increase the performance, we will use different thresholds δ_j on different scales j , that are of the form

$$\delta_j = K_j \sigma$$

where for four scales, we typically have $K = [K_j]_j = [2.5, 2.5, 2.5, 3.8]$.

The quality of the reconstruction is measured using the peak signal-to-noise ratio (PSNR), defined as

$$PSNR = 20 \log_{10} \frac{255\sqrt{N}}{\|f - f_{denoised}\|_F} \quad (35)$$

where N is the number of pixels and $\|\cdot\|_F$ denotes the Frobenius norm. Note that 255 is the maximum value a pixel can attain in a grayscale image.

We compare the performance of $SL2D_1$, $SL2D_2$, the nonsubsampling shearlet transform $NSST$, the nonsubsampling contourlet transform $NSCT$, the fast discrete curvelet transform $FDCT$, and the stationary wavelet transform SWT . All quantitative results for a total of four grayscale images and several levels of noise as well as the running times and redundancies associated with the considered transforms are displayed in Figure 11. For a visual comparison, we refer to Figure 19 in the Appendix.

5.4. Image Inpainting

We consider a grayscale image $f \in \ell^2(\mathbb{Z}^2)$ to be partially occluded by a binary mask $M \in \{0, 1\}^{\mathbb{Z} \times \mathbb{Z}}$, i.e.,

$$f_{masked}(i, j) = f(i, j)M(i, j).$$

The algorithm for inpainting the missing parts is based on an iterative thresholding scheme published in 2005 by Starck et al. [Starck et al. 2005] (see also [Fadili et al. 2010] and Algorithm 3). In each step, a forward transform is performed on the unoccluded parts of the image combined with everything already inpainted in the missing areas. The resulting coefficients are then subject to hard thresholding before an inverse transform is carried out. By gradually decreasing the thresholding constant, this algorithm approximates a sparse set of coefficients whose synthesis is very close to the original image on the unoccluded parts.

Again, we use the PSNR defined in (35) to measure the qualitative performance and to compare the systems $SL2D_1$ and $SL2D_2$ with the nonsubsampling shearlet transform ($NSST$), the fast discrete curvelet transform ($FDCT$) and the stationary wavelet transform (SWT). To maximize the quality of the output, we perform 300 iterations during each inpainting task. Due to this heavy computational workload, we do not consider the nonsubsampling contourlet transform ($NSCT$), as one iteration would already take more than 7 minutes. Furthermore, we use three different types of masks to properly emulate several typical inpainting problems, which are displayed in Figure 12.

| Image Denoising: Running Times | | | |
|--------------------------------|------------|--------------|------------------------|
| | Redundancy | Running Time | Running Time with CUDA |
| $SL2D_1$ | 25 | 4.8 s | 1.7 s |
| $SL2D_2$ | 49 | 10.9 s | 5.7 s |
| $NSST$ | 49 | 9.4 s | - |
| $NSCT$ | 49 | 452 s | - |
| $FDCT$ | 2.85 | 0.8 s | - |
| SWT | 13 | 0.7 s | - |

| Image Denoising: Quantitative Results in PSNR | | | | | | | | | | |
|---|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Lenna | | | | | Barbara | | | | |
| | $\sigma=10$ | 20 | 30 | 40 | 50 | 10 | 20 | 30 | 40 | 50 |
| $SL2D_1$ | 35.79 | 32.51 | 30.52 | 29.01 | 27.79 | 33.38 | 29.42 | 27.03 | 25.40 | 24.37 |
| $SL2D_2$ | 35.90 | 32.75 | 30.87 | 29.43 | 28.26 | 33.63 | 29.98 | 27.83 | 26.28 | 25.17 |
| $NSST$ | 35.85 | 32.83 | 31.06 | 29.69 | 28.58 | 33.56 | 29.91 | 27.75 | 26.20 | 25.15 |
| $NSCT$ | 35.67 | 32.53 | 30.69 | 29.30 | 28.14 | 33.43 | 29.58 | 27.36 | 25.89 | 24.88 |
| $FDCT$ | 34.01 | 31.41 | 29.71 | 28.38 | 27.33 | 28.96 | 25.47 | 24.48 | 23.87 | 23.43 |
| SWT | 34.19 | 30.79 | 28.91 | 27.64 | 26.62 | 31.08 | 26.55 | 24.55 | 23.55 | 23.00 |

| | Boat | | | | | Peppers | | | | |
|----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | $\sigma=10$ | 20 | 30 | 40 | 50 | 10 | 20 | 30 | 40 | 50 |
| $SL2D_1$ | 33.06 | 30.00 | 28.16 | 26.87 | 25.86 | 34.10 | 31.78 | 30.04 | 28.67 | 27.49 |
| $SL2D_2$ | 33.14 | 30.18 | 28.42 | 27.17 | 26.18 | 34.12 | 31.92 | 30.32 | 29.06 | 27.97 |
| $NSST$ | 33.05 | 30.09 | 28.34 | 27.14 | 26.20 | 34.05 | 31.84 | 30.26 | 29.05 | 28.02 |
| $NSCT$ | 32.87 | 29.82 | 28.05 | 26.83 | 25.90 | 33.87 | 31.51 | 29.83 | 28.53 | 27.51 |
| $FDCT$ | 30.73 | 28.36 | 27.00 | 25.99 | 25.16 | 32.37 | 29.67 | 28.17 | 27.11 | 26.25 |
| SWT | 31.54 | 28.23 | 26.45 | 25.31 | 24.47 | 33.09 | 30.39 | 28.44 | 26.99 | 25.92 |

Fig. 11. Numerical results of the image denoising experiment.

ALGORITHM 3: Image inpainting via iterative thresholding**Data:** $f_{\text{masked}}, M, \delta_{\text{init}}, \delta_{\text{min}}, \text{iterations}$ **Result:** $f_{\text{inpainted}}$ $f_{\text{inpainted}} := 0;$ $\delta := \delta_{\text{init}};$ $\lambda := (\delta_{\text{min}})^{1/(\text{iterations}-1)};$ **for** $i := 1$ **to** iterations **do** $f_{\text{res}} := M \cdot (f_{\text{masked}} - f_{\text{inpainted}});$ // \cdot denotes the pointwise multiplication $f_{\text{inpainted}} := \mathcal{T}^{-1} \mathcal{T}_{\delta} \mathcal{T}(f_{\text{res}} + f_{\text{inpainted}});$ // Forward transform, thresholding, synthesis $\delta := \lambda \delta;$ **end**

For the quantitative results and a compilation of running times and redundancies see Figure 13. A visual comparison is provided in Figure 20 in the Appendix.

5.5. Image Decomposition

We assume a given image $f \in \ell^2(\mathbb{Z}^2)$ can be split into a curvilinear part f_0 and a part f_1 containing isotropic blob-like structures in the sense of

$$f = f_0 + f_1. \quad (36)$$

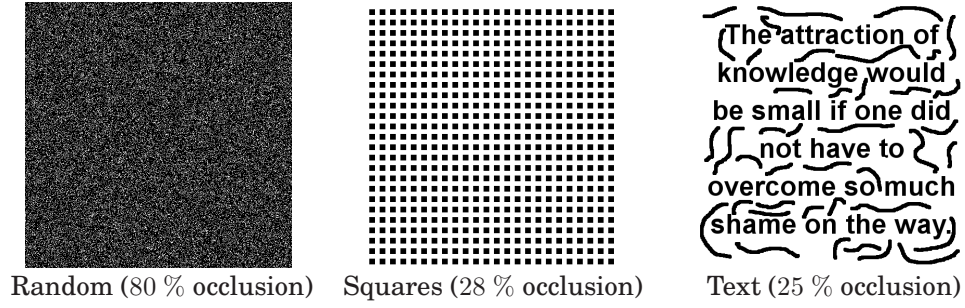


Fig. 12. Three different binary masks.

| Image Inpainting: Running Times | | | |
|---------------------------------|------------|--------------|------------------------|
| | Redundancy | Running Time | Running Time with CUDA |
| $SL2D_1$ | 25 | 463 s | 45.3 s |
| $SL2D_2$ | 49 | 931 s | 78.7 s |
| $NSST$ | 49 | 2495 s | - |
| $FDCT$ | 2.85 | 193 s | - |
| SWT | 13 | 162 s | - |

| Image Inpainting: Quantitative Results in PSNR | | | | | | | | | |
|--|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Lenna | | | Barbara | | | Flintstones | | |
| | Rand | Squares | Text | Rand | Squares | Text | Rand | Squares | Text |
| $SL2D_1$ | 32.17 | 33.11 | 31.44 | 26.59 | 30.08 | 29.70 | 23.84 | 24.16 | 22.98 |
| $SL2D_2$ | 32.08 | 33.92 | 32.57 | 27.82 | 31.53 | 31.05 | 23.59 | 25.14 | 23.72 |
| $NSST$ | 32.31 | 33.79 | 32.78 | 27.62 | 31.23 | 30.93 | 24.12 | 24.89 | 23.57 |
| $FDCT$ | 30.40 | 32.78 | 32.06 | 24.07 | 27.78 | 28.00 | 22.08 | 23.63 | 22.78 |
| SWT | 30.46 | 31.69 | 30.95 | 23.84 | 28.46 | 28.75 | 21.86 | 22.91 | 22.18 |

Fig. 13. Numerical results of the image inpainting experiment.

To compute a meaningful decomposition, we use a directional transform nicely adapted to curvilinear structures (e.g., shearlets or curvelets) together with the stationary wavelet transform, and apply the iterative thresholding algorithm from [Starck et al. 2005] (see also [Fadili et al. 2010] and Algorithm 4). During each iteration, the difference between the input and the current blob-like image is subject to a directional transform, while the difference between the input and the current curvilinear image is subject to a stationary wavelet transform. Before computing the inverse transforms, a hard thresholding operator is applied to both sets of coefficients. By gradually decreasing the thresholding constant, this algorithm iteratively approximates an image close to the curvilinear part of the original data whose coefficients are sparse in the directional dictionary and an image close to the blob-like part of the original input whose coefficients are sparse with respect to the stationary wavelet transform. An example can be seen in Figure 14.

To quantitatively measure the performance of a sparse representation scheme, we restrict our experiments to binary images as in Figure 14. Furthermore, we introduce the operator B_δ , mapping images $g \in \ell^2(\mathbb{Z}^2)$ to binary images with

$$(B_\delta g)(i, j) = \begin{cases} 1 & \text{if } |g(i, j)| \geq \delta, \\ 0 & \text{else,} \end{cases}$$

ALGORITHM 4: Image decomposition via iterative thresholding**Data:** $f, \delta_{init}, \delta_{min}, iterations$ **Result:** \tilde{f}_0, \tilde{f}_1 $\tilde{f}_0 := \tilde{f}_1 := 0;$ $\delta := \delta_{init};$ $\lambda := (\delta_{min})^{1/(iterations-1)};$ **for** $i := 1$ **to** $iterations$ **do** $f_{res} := f - (f_0 + f_1);$ $\tilde{f}_0 := \mathcal{T}_0^{-1} T_\delta \mathcal{T}_0(f_{res} + \tilde{f}_0);$ // Hard thresholding with anisotropic transform $\tilde{f}_1 := \mathcal{T}_1^{-1} T_\delta \mathcal{T}_1(f_{res} + \tilde{f}_1);$ // Hard thresholding with isotropic transform

// To improve the quality of the result, you can add a total variation

correction for the directional part

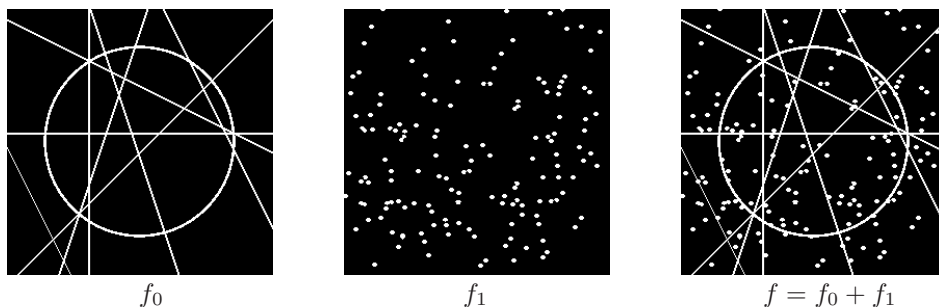
 $\delta := \lambda \delta;$ **end**

Fig. 14. In the image decomposition experiment, we try to recover the curvilinear part f_0 and the isotropic part f_1 from $f = f_0 + f_1$.

and the measures (see [Kutyniok and Lim 2012])

$$Q(\tilde{f}_0, \delta) = \frac{\|g * f_0 - g * (B_\delta \tilde{f}_0)\|_2}{\|g * f_0\|_2},$$

$$Q_{opt}(\tilde{f}_0) = \min_{\delta \in \{0, \dots, 255\}} Q(\tilde{f}_0, \delta),$$

where \tilde{f}_0 is the curvilinear part computed by algorithm 3, g is a discrete two-dimensional Gaussian filter and $*$ denotes the convolution. This definition can naturally also be used for the blob-like part \tilde{f}_1 .

As input for our experiment, we used the image depicted in Figure 14. We compared the systems $SL2D_1$ and $SL2D_2$ to the directional transforms $NSST$ and $FDCT$. The numerical results together with redundancies and running times are compiled in Figures 15 and 16. For a visual comparison of the results, see Figure 21 in the Appendix.

5.6. Video Denoising

Similar to the two-dimensional case, we consider grayscale videos of size 192x192x192 distorted with Gaussian white noise. For a video $f \in \ell^2(\mathbb{Z}^3)$, we have

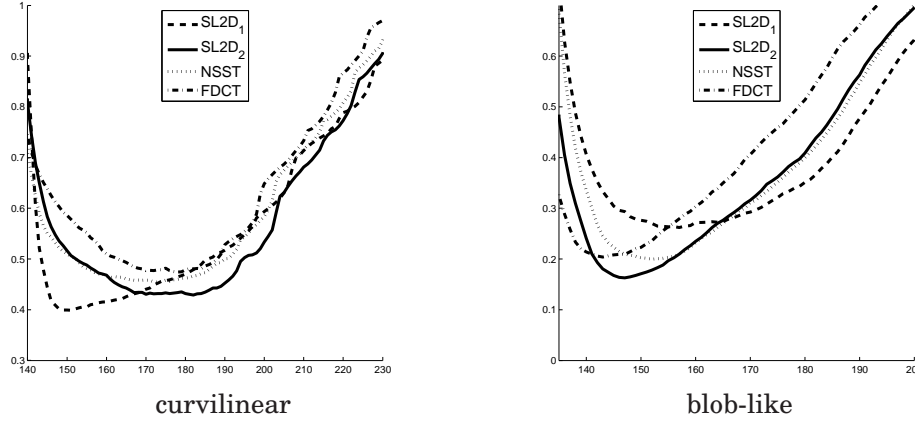
$$f_{noisy}(i, j, k) = f(i, j, k) + e(i, j, k)$$

where $e(i, j, k) \sim \mathcal{N}(0, \sigma^2)$, and compute

$$f_{denoised} = \mathcal{T}^{-1} T_\delta \mathcal{T} f_{noisy}$$

| Image Decomposition: Quantitative Results and Running Times | | | | |
|--|-----------------------------|-----------------------------|------------|--------------|
| | Points | Curves | Redundancy | Running Time |
| $SL2D_1$ | $Q_{opt} = \mathbf{0.3991}$ | $Q_{opt} = 0.2620$ | 25 | 57.1 s |
| $SL2D_2$ | $Q_{opt} = 0.4288$ | $Q_{opt} = \mathbf{0.1627}$ | 49 | 87.8 s |
| $NSST$ | $Q_{opt} = 0.4541$ | $Q_{opt} = 0.1996$ | 49 | 203.7 s |
| $FDCT$ | $Q_{opt} = 0.4743$ | $Q_{opt} = 0.2040$ | 2.85 | 84.9 s |

Fig. 15. Numerical results of the image decomposition experiment.


 Fig. 16. The values $Q(\tilde{f}_0, \delta)$ and $Q(\tilde{f}_1, \delta)$ as functions of δ .

where \mathcal{T} denotes the forward and \mathcal{T}^{-1} the inverse transform associated with a sparse approximation scheme and T_δ is the hard thresholding operator already defined in (34). Again, our thresholds will be of the form

$$\delta_j = K_j \sigma,$$

where j iterates the scales of the digital transform. Typically, we will use systems with three scales and choose $K = [K_j]_j = [3, 3, 4]$.

In total, we run our experiment with three different videos and noise levels ranging from $\sigma = 10$ to 50. To quantitatively compare the performance of ShearLab 3D with the performance of the three-dimensional nonsubsampling shearlet transform ($NSST$) and the surfacelet transform, we again calculate the peak signal to noise ratio (PSNR) defined in (35). For a complete listing of our numerical results, see Figure 17. A visual comparison is provided in Figure 22 in the Appendix.

5.7. Video Inpainting

Analogous to the two-dimensional case, we consider grayscale videos $f \in \ell^2(\mathbb{Z}^3)$ of size $192 \times 192 \times 192$ to be partially occluded by a binary mask $M \in \{0, 1\}^{\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}}$, i.e.,

$$f_{masked}(i, j, k) = f(i, j, k)M(i, j, k)$$

To fill in the missing gaps, we run Algorithm 3 (see [Starck et al. 2005] and [Fadili et al. 2010]) with 50 iterations.

In our video inpainting experiments, we use a random mask with 80 % occlusion and a mask consisting of cubes of random size with 5 % occlusion. We compare the systems $SL3D_1$ and $SL3D_2$ to the surfacelet transform $SURF$. All numerical results are compiled in Figure 18, and for a visual comparison we refer to Figure 23 in the Appendix.

| Video Denoising: Quantitative Results in PSNR and Running Times | | | | | | | | | | |
|---|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Mobile | | | | | Coastguard | | | | |
| | $\sigma = 10$ | 20 | 30 | 40 | 50 | 10 | 20 | 30 | 40 | 50 |
| $SL3D_1$ | 35.27 | 31.32 | 29.01 | 27.38 | 26.14 | 33.13 | 29.46 | 27.51 | 26.17 | 25.18 |
| $SL3D_2$ | 35.91 | 32.19 | 29.99 | 28.43 | 27.23 | 33.81 | 30.28 | 28.41 | 27.14 | 26.17 |
| $NSST$ | 34.61 | 30.83 | 28.56 | 26.96 | 25.73 | 32.59 | 29.00 | 27.05 | 25.68 | 24.63 |
| $SURF$ | 32.79 | 29.96 | 28.25 | 27.04 | 26.11 | 30.86 | 28.26 | 26.87 | 25.91 | 25.18 |
| $SL2D_2$ | 32.49 | 28.58 | 26.42 | 24.97 | 23.90 | 31.24 | 27.67 | 25.79 | 24.52 | 23.59 |

| | Tennis | | | | | Redundancy | Running Time | Running Time with CUDA |
|----------|--------------|--------------|--------------|--------------|--------------|------------|--------------|------------------------|
| | 10 | 20 | 30 | 40 | 50 | | | |
| $SL3D_1$ | 33.76 | 30.19 | 28.28 | 26.94 | 25.90 | 76 | 146 s | 14.1 s |
| $SL3D_2$ | 34.15 | 30.67 | 28.86 | 27.62 | 26.67 | 292 | 610 s | 53.1 s |
| $NSST$ | 33.02 | 29.48 | 27.55 | 26.20 | 25.14 | 208 | 512 s | - |
| $SURF$ | 29.95 | 27.34 | 26.02 | 25.11 | 24.43 | 6.4 | 33 s | - |
| $SL2D_2$ | 30.91 | 27.48 | 25.55 | 24.31 | 23.42 | 49 | 43 s | 26.4 s |

Fig. 17. Numerical results of the video denoising experiment.

| Video Inpainting: Quantitative Results in PSNR and Running Times | | | | | | | |
|--|--------------|--------------|--------------|--------------|------------|--------------|------------------------|
| | Mobile | | Coastguard | | Redundancy | Running Time | Running Time with CUDA |
| | Rand | Squares | Rand | Squares | | | |
| $SL3D_1$ | 28.15 | 31.65 | 26.72 | 29.05 | 76 | - | 837 s |
| $SL3D_2$ | 29.98 | 32.87 | 28.09 | 30.22 | 292 | - | 3101 s |
| $SURF$ | 22.40 | 28.24 | 21.50 | 27.1 | 6.4 | 852 s | - |
| $SL2D_2$ | 23.63 | 31.61 | 23.42 | 30.80 | 49 | - | 1350 s |

Fig. 18. Numerical results of the video inpainting experiment.

5.8. Discussion

In all experiments, the sparse approximations provided by ShearLab 3D yield the best results with respect to the applied quantitative measures, except for some cases where our algorithm is slightly outperformed by the nonsubsampling shearlet transform ($NSST$) (see, for example, Figure 11). However, the $NSST$ has significantly worse running times in most experiments, in particular, if a CUDA capable GPU is available (cf, e.g., Figure 13). In computationally heavy tasks like image or video inpainting, applying CUDA can lead to a significantly increased speed (up to a factor 10 in Figures 13 and 17).

The main goal of our experiments was not to argue that the digital shearlet transform implemented in ShearLab 3D is specifically adapted to a certain task like image denoising, but to compare its applicability to other, similar transforms. That being said, we would like to mention that our video denoising results are only marginally beaten by the BM3D algorithm [Maggioni et al. 2012] which represents – to our knowledge – the current state of the art (PSNR values for the coastguard sequence, $144 \times 176 \times 300$, $\sigma = 30$, BM3D: 29.69, $SL3D_2$: 29.54).

REFERENCES

- Roberto H. Bamberger and Mark J. T. Smith. 1992. A Filter Bank for the Directional Decomposition of Image: Theory and Design. *IEEE Trans. Image Proc.* 40 (1992), 882–893.
- Emmanuel Candès, Laurent Demanet, David L. Donoho, and Lexing Ying. 2006. Fast Discrete Curvelet Transform. *SIAM Multiscale Model. Simul.* 5 (2006), 861–899.
- Emmanuel Candès and David L. Donoho. 1999a. Curvelets – a surprisingly effective nonadaptive representation for objects with edges. *Curves and Surfaces* (1999).

- Emmanuel Candès and David L. Donoho. 1999b. Ridgelets: a key to higher-dimensional intermittency?., *Phil. Trans. R. Soc. Lond. A*. 357 (1999), 2495–2509.
- Emmanuel Candès and David L. Donoho. 2004. New Tight Frames of Curvelets and Optimal Representation of Objects with C^2 Singularities. *Comm. Pure. Appl. Math.* 57 (2004), 219–266.
- O. Christensen. 2003. *An Introduction to Frames and Riesz Bases*. Birkhäuser, Boston.
- Albert Cohen, Ingrid Daubechies, and Jean-Christophe Feauveau. 1992. Biorthogonal Bases of Compactly Supported Wavelets. *Communications on Pure and Applied Mathematics* 45 (1992), 1992.
- Arthur L. da Cunha, Jianping Zhou, and Minh N. Do. 2006. The Nonsubsampled Contourlet Transform: Theory, Design and Applications. *IEEE Trans. Image Proc.* 15 (2006), 3089–3101.
- Ingrid Daubechies. 1992. *Ten Lectures on Wavelets*. SIAM, Philadelphia.
- Mark Davenport, Marco Duarte, Yonina Eldar, and Gitta Kutyniok. 2012. *Compressed Sensing: Theory and Applications*. Cambridge University Press, Chapter Introduction to Compressed Sensing, 1–68.
- Minh N. Do and Yue M. Lu. 2007. Multidimensional Directional Filter Banks and Surfacelets. *IEEE Trans. Image Proc.* 16 (2007), 918–931.
- Minh N. Do and Martin Vetterli. 2005. The Contourlet Transform: An Efficient Directional Multiresolution Image Representation. *IEEE Trans. Image Proc.* 14 (2005), 2091–2106.
- David L. Donoho, Arian Maleki, Morteza Shahram, Victoria Stodden, and Inam Ur-Rahman. 2009. Fifteen years of reproducible research in computational harmonic analysis. *Comput. Sci. Engrg.* 11 (2009), 8–18.
- Glenn Easley, Demetrio Labate, and Wang-Q Lim. 2008. Sparse Directional Image Representation using the Discrete Shearlet Transform. *Appl. Comput. Harmon. Anal.* 25 (2008), 25–46.
- Jalal M. Fadili, Jean-Luc Starck, Michael Elad, and David. 2010. MCALab: Reproducible Research in Signal and Image Decomposition and inpainting. *Comput. Sci. Eng.* 12 (2010), 44–63.
- Martin Genzel and Gitta Kutyniok. 2014. Asymptotical Analysis of Inpainting via Universal Shearlet Systems. *preprint* (2014).
- K. Gröchenig. 1993. Acceleration of the frame algorithm. *IEEE Trans. Signal Process.* 41 (1993), 3331–3334.
- Philipp Grohs, Sandra Keiper, Gitta Kutyniok, and Martin Schäfer. 2013. α -Molecules. *preprint* (2013).
- Philipp Grohs and Gitta Kutyniok. 2014. Parabolic Molecules. *Found. Comput. Math.* (2014). to appear.
- Kanghui Guo, Gitta Kutyniok, and Demetrio Labate. 2006. Sparse multidimensional representations using anisotropic dilation and shear operators. In *Wavelets and Splines (Athens, GA, 2005)*. Nashboro Press, Nashville, TN, 189–201.
- Kanghui Guo and Demetrio Labate. 2012. Optimally sparse representations of 3D data with C^2 surface singularities using Parseval frames of shearlets. *SIAM J. Math. Anal.* 44 (2012), 851–886.
- Bin Han, Gitta Kutyniok, and Zouwei Shen. 2011. Adaptive Multiresolution Analysis Structures and Shearlet Systems. *SIAM J. Numer. Anal.* 49 (2011), 1921–1946.
- Felix Herrmann, Deli Wang, Gilles Hennenfent, and Peyman P. Moghaddam. 2008. Curvelet-based seismic data processing: a multiscale and nonlinear approach. *Geophysics* 73 (2008), A1–A5.
- Sandra Keiper. 2012. *A Flexible Shearlet Transform - Sparse Approximations and Dictionary Learning*. Ph.D. Dissertation. Technische Universität Berlin.
- Pisamai Kittipoom, Gitta Kutyniok, and Wang-Q Lim. 2012. Construction of Compactly Supported Shearlet Frames. *Constr. Approx.* 35 (2012), 21–72.
- Gitta Kutyniok and Demetrio Labate. 2012. *Shearlets: Multiscale Analysis for Multivariate Data*. Birkhäuser Boston, Chapter Introduction to Shearlets, 1–38.
- Gitta Kutyniok, Jakob Lemvig, and Wang-Q Lim. 2012a. Optimally Sparse Approximations of 3D Functions by Compactly Supported Shearlet Frames. *SIAM J. Math. Anal.* 44 (2012), 2962–3017.
- Gitta Kutyniok and Wang-Q Lim. 2011. Compactly Supported Shearlets are Optimally Sparse. *J. Approx. Theory* 163 (2011), 1564–1589.
- Gitta Kutyniok and Wang-Q Lim. 2012. Image Separation using Wavelets and Shearlets. *Lecture Notes in Computer Science* 6920 (2012), 416–430.
- Gitta Kutyniok and Wang-Q Lim. 2013. Optimal sparse sampling for Fourier data. *preprint* (2013).
- Gita Kutyniok and Tomas Sauer. 2009. Adaptive Directional Subdivision Schemes and Shearlet Multiresolution Analysis. *SIAM J. Math. Anal.* 41 (2009), 1436–1471.
- Gitta Kutyniok, Morteza Shahram, and Xiaosheng Zhuang. 2012b. ShearLab: A Rational Design of a Digital Parabolic Scaling Algorithm. *SIAM J. Imaging Sci.* 5 (2012), 1291–1332.
- Wang-Q Lim. 2010. The discrete shearlet transform: A new directional transform and compactly supported shearlet frames. *IEEE Trans. Image Proc.* 19 (2010), 1166–1180.
- Wang-Q Lim. 2013. Nonseparable shearlet transform. *IEEE Trans. Image Proc.* (2013). to appear.

- Mattea Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. 2012. Video Denoising, Deblocking and Enhancement Through Separable 4-D Nonlocal Spatiotemporal Transforms. *IEEE Trans. Image Proc.* 21 (2012), 3952–3966.
- Stéphane Mallat. 2008. *A Wavelet Tour of Signal Processing: The Sparse Way* (2nd ed.). Academic Press.
- Pooran Singh Negi and Demetrio Labate. 2012. 3-D Discrete Shearlet Transform and Video Processing. *IEEE Trans. Image Proc.* 21 (2012), 2944–2954.
- Eero Simoncelli, William Freeman, Edward H. Adelson, and David Heeger. 1992. Shiftable multi-scale transforms. *IEEE Trans Information Theory* 38(2) (1992), 587–607.
- Jean-Luc Starck, Michael Elad, and David L. Donoho. 2005. Image Decomposition via the Combination of Sparse Representation and a Variational Approach. *IEEE Transactions on Image Processing* 14 (2005), 1570–1582.
- Jean-Luc Starck, Fionn Murtagh, and Jalal Fadili. 2010. *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity*. Cambridge University Press, Cambridge.



Fig. 19. The Barbara image is distorted with Gaussian white noise with $\sigma = 40$ and denoised using various sparse approximation schemes.



Fig. 20. The Lenna image is occluded with a random binary mask and denoised using various sparse approximation schemes.

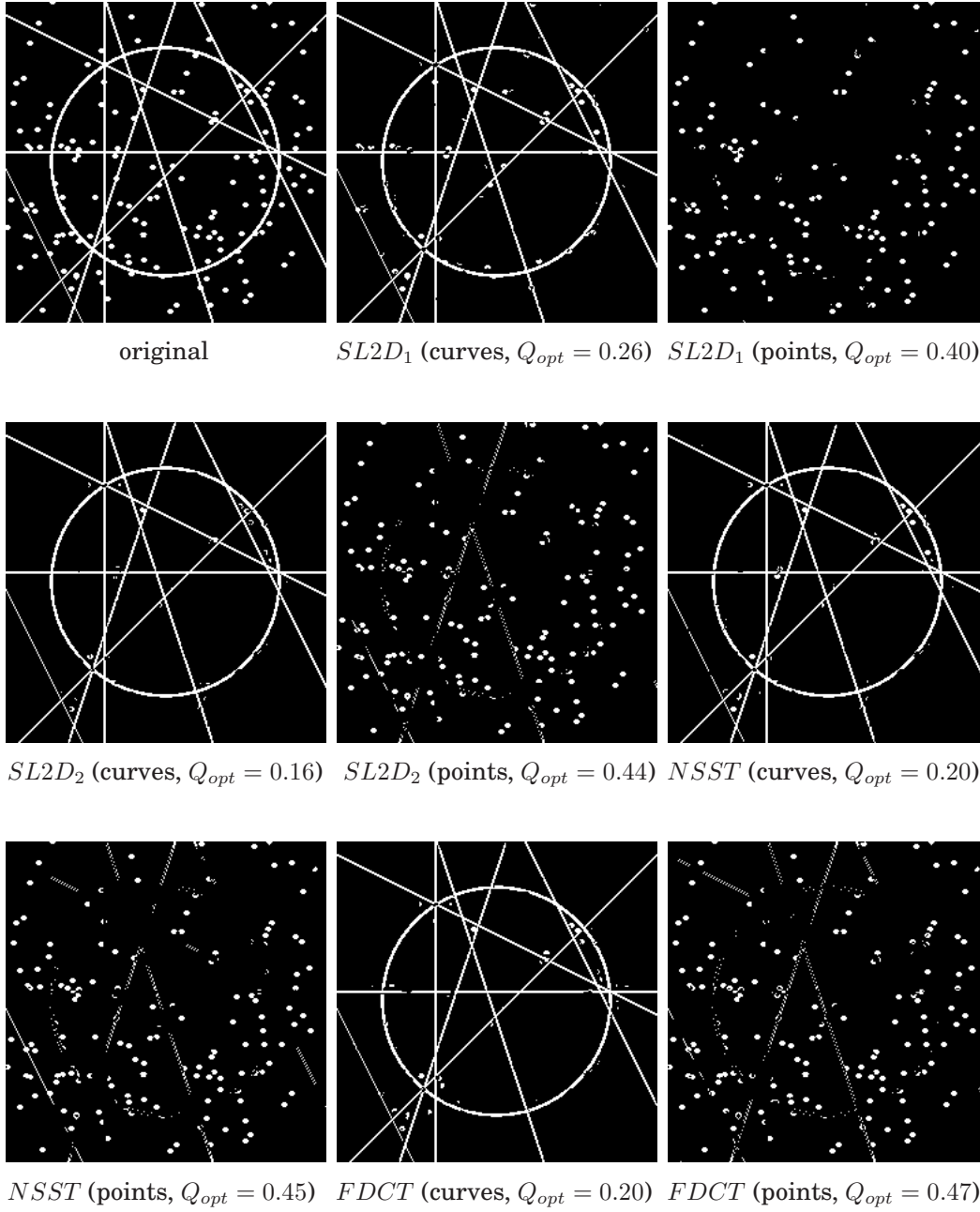


Fig. 21. The optimal decompositions of a binary image obtained from applying algorithm 4 using four different directional transforms.



Fig. 22. The original, noisy and denoised frame 110 of the coastguard sequence..

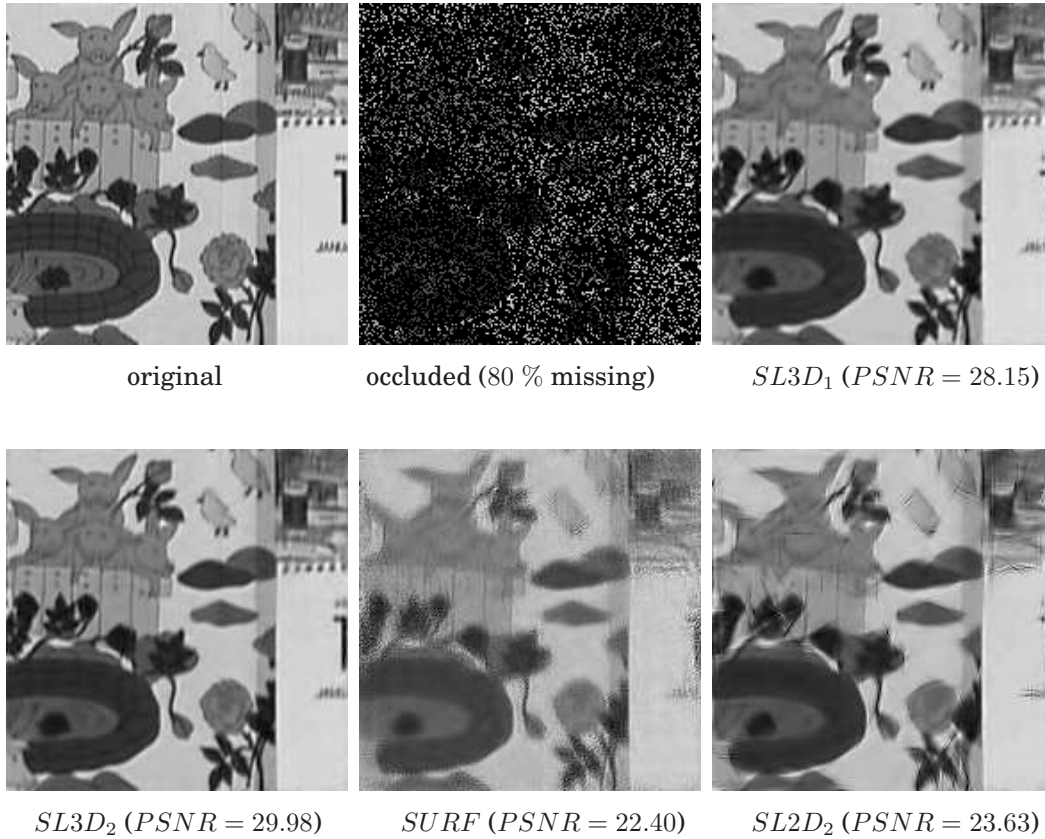


Fig. 23. The original, occluded and inpainted frame 37 of the mobile sequence.